Atlas 1.1: An Update to the Theory of Effective Systems Engineers

SERC-2018-TR-101-A

January 16, 2018

Principal Investigator: Dr. Nicole A.C. Hutchison, Stevens Institute of Technology

Co-Principal Investigator: Dr. Dinesh Verma, Stevens Institute of Technology

Research Team:

Stevens Institute of Technology: Dr. Pamela Burke, Ms. Megan Clifford, Mr. Ralph Giffin, Mr. Sergio Luna, Mr. Matthew Partacz

Sponsor: Office of the Deputy Assistant Secretary of Defense for Systems Engineering DASD(SE)

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract H98230-08-D-0171 and HQ0034-13-D-0004.

TABLE OF CONTENTS

List of Figures

List of Tables

ACKNOWLEDGEMENTS

The Helix team would like to thank all the organizations and individuals that participated in the project, offering their resources, time, and effort. This was critical to our research. Their active participation in the Helix interviews provided us data that was rich in both quality and quantity, which makes this research more valuable and useful to the participating organizations and the systems engineering community at large. To the organizations that have publicly shared information about their experiences with using *Atlas*, we owe tremendous thanks; without them, this document would not be possible.

We are most grateful to the Office of the Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)), especially Kristen Baldwin and Scott Lucero, for their continued support, without which this research would not be possible. The International Council on Systems Engineering (INCOSE) and the National Defense Industrial Association Systems Engineering Division (NDIA-SED) are both valued partners in this research and we thank them, especially Courtney Wright, David Long, Bill Miller, and Don Gelosh.

We thank all former members of the Helix research team whose contributions have shaped our research over the years. In particular, we would like to thank Dr. Art Pyster, whose leadership and dedication since the beginning of the project were critical and instrumental to its success and who still champions and supports Helix.

I would also like to personally thank the current Helix team, who have helped tremendously to mature not only the project, but my own thinking and views on it. Pam, Ralph, Sergio, Megan, Matthew, and Dinesh, please accept my gratitude for your hard work and dedication.

Nicole AC Hutchison

Nicole AC Hutchison
Helix Principal Investigator

EXECUTIVE SUMMARY

In 2017, the Helix team conducted additional work on effective systems engineering capabilities, culminating in *Atlas* 1.1, which is an incremental update reflecting additional analysis of existing data as well as additional data collection in 2018. Though the changes in *Atlas 1.1* are relatively minor (as reflected in the ".1" version number), they nevertheless reflect not additional data collection and analyses, but also incorporate feedback from the community. The Helix team presented their work at several community events, including the IISE annual conference, the INCOSE International Symposium, the NDIA Systems Engineering Conference, and the SERC Sponsored Research Review (SSRR). At each of these events, the team gained feedback from the community, collecting frequently asked questions, uncovering areas of confusion, and identifying areas for improvement. The changes include:

- Reordering of the values systems engineers provide to reflect the frequency at which they occurred in the dataset along with minor cleanup of the value names;

- Updating the "Requirements Owner" and "Systems Architect" roles. The activities around functional architecture were moved from Requirements Owner to Systems Architect which both better reflect the realities of the grouping of these activities in practice, but are groupings which better align with the mental models of most individuals who have engaged with the Helix team in 2017.

- There were several minor edits to the proficiency model. The proficiency areas stayed the same, though the area formerly titles "Systems Engineering Mindset" is now "Systems Mindset". Within this area, the category formerly titled "flexibility" has been renamed "adaptability". This not only better reflects the comments in the Helix interviews – which revolved around the ability of an individual to cope with a change – but also reduces confusion. The distinction between proficiencies and personal enabling characteristics is nuanced, and the term "flexibility" caused confusion about the classification of the category. In addition, the titles of categories in the "Technical Leadership" proficiency area were updated to increase clarity. The previous titles implied overlap; e.g. "Managing Stakeholders with Diverse and Conflicting Needs" and "Conflict Resolution and Barrier Breaking" seemed to overlap, though their topics were different. Though they are related, they are distinct. The Helix team renamed "Managing Stakeholder with Diverse and Conflicting Needs" to "Managing Diverse Stakeholders".

- Personal enabling characteristics were updated with minor changes in the definitions.

With these changes, the Helix team has reflected all it has learned from additional data collection and supporting organizations that are implementing *Atlas* in 2017.

1 THE NEED FOR *ATLAS* 1.1

In December 2016, the Helix team published *Atlas* 1.0. This remains a milestone that the team is very proud of. However, in the spirit of continual growth and development, in the year since publication, the Helix team has gathered feedback from the community on how to improve *Atlas*. This document presents incremental improvements in *Atlas*, with the version number – 1.1. – representing the minor improvements. Based on the feedback in 2017, there is no need for a major evolutionary change to the approach.

This is a comprehensive update to *Atlas* – the changes are highlighted in the text, but information unchanged from 1.0 is also included making this a stand-alone document that should replace version 1.0 in use.

1.1 CHANGES SINCE 1.0

Most of the changes in *Atlas* 1.1 are around improvements of language, particularly in the way elements are titled. As the team was reminded this year, words matter and sometimes seemingly small changes can create large leaps in common understanding.

One of the major aspects reviewed for *Atlas* 1.1 was the relationship between "Proficiency" and "Personal Characteristics". When presented at community events, there were commonly questions about the overlap between the two. Proficiencies are knowledge, skills, abilities, behaviors, and cognitions that an individual utilizes to perform systems engineering. There are clear ways to grow proficiencies and individuals can be guided on growth paths for these. Think of something like "lifecycle" – an individual may take a graduate course on lifecycle models and be guided to work on multiple projects in different development phases to see the whole lifecycle and, likely, this person would then become better at understanding system lifecycles. Personal characteristics, however, are more internally focused and much more difficult to grow. This does not mean that an individual can not improve. Take, for example, the personal characteristic of "self-awareness". An individual can be told that self awareness is important, given tools to improve self-awareness, and participate in 360° feedback to give them information to improve their self-awareness. Some individuals will internalize this and become markedly more self aware; others receiving the same information may not change in self awareness at all. While this is true to some extent for anything, the Helix team views proficiencies as skills that are more easily influenced externally versus personal characteristics, which are largely dependent on internal factors. The team reviewed the Proficiencies and Personal Characteristics for *Atlas* and has made some minor adjustments to improve the crispness and distinction between the two.

The following is a summary of the changes found in *Atlas* 1.1:

- Minor updates to the values including cleanup of the value names and reordering.
- Minor updates to proficiency model, particularly in terms of proficiency names to improve clarity and reduce duplication.
- Minor updates to the organization of the systems engineering roles found in Helix to reflect community feedback.

- Minor updates to personal characteristics, particularly in terms of how personal characteristics may be addressed.

In addition to the changes in 1.1, the previous version contained sections called "Implications for Use." The Helix team has created a companion *Atlas 1.1 Implementation Guide*, (SERC-2018-TR-101-B). Insights previously found in "Implications for Use" sections have been moved to the *Implementation Guide*.

1.2 How is *Atlas* Different from Helix?

Helix is the name of the overarching SERC project. Helix has been examining what makes systems engineers effective for over four years. As a project, Helix has created many different deliverables or products. The primary product of Helix is *Atlas: The Theory of Effective Systems Engineers*. This document represents *Atlas 1.1* – expected to be mature enough for individuals or organizations to use without direct help from the Helix team. It is a standalone document to detail the contents of *Atlas*.
This document does *not* contain all of the research that led to the development of *Atlas 1.1.* Instead, the detailed research results and how they led to *Atlas 1.0* are contained in the companion Helix Technical Report (SERC-2018-TR-101). Individuals or organizations that want not just to use *Atlas* but to also understand the rationale and methodology behind its development should reference the *Technical Report*. Several earlier published Helix papers and technical reports are also referred to throughout this report. The reader is not expected to read the earlier technical reports or any of the other Helix papers or reports, in order to understand *Atlas 1.1*.

In addition to the *Technical Report*, in 2017-8 the Helix team generated the *Atlas Career Path Guidebook* – this document provides analyses of the Helix dataset, providing common patterns in systems engineers' careers. The *Guidebook* also provides some insights on questions commonly asked of the Helix team around career paths and the team's responses. Finally, additional work on linking proficiencies to career paths has been completed and is reflected in the *Guidebook*. (SERC-2018-TR-101-C) The team also generated the *Atlas 1.1 Implementation Guide* – Whenever *Atlas* is presented, there are many questions about how to take the theory and apply it in practice. The Guide provides examples from organizations that have implemented parts of *Atlas*, and guidance created by the Helix team based on many interactions with organizations around implementation as well as the extensive Helix dataset. (SERC-2018-TR-101-B)

There are tools that an individual or organization can use to support self-assessment using *Atlas*. The paper-based tools are contained in the Appendices of this report. The team has also developed more easily tailored Excel-based tools, which can be found on the Helix page of the SERC website (http://www.sercuarc.org/projects/helix/).

The relationship between Helix, *Atlas*, the Technical Reports, and the tools is illustrated in Figure 1.

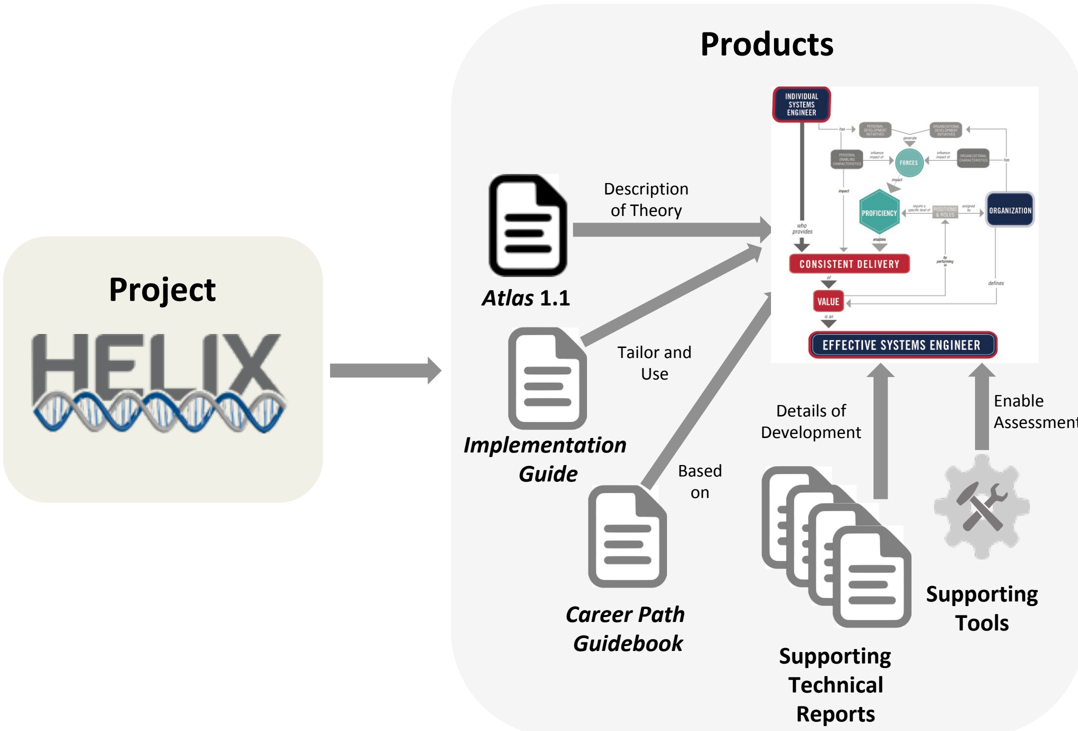

Figure 1. Relationship between Helix and Atlas

1.3 INCREMENTAL *ATLAS* DEVELOPMENT

The Helix project used an incremental approach to develop Atlas. This approach was designed to enable publication and use of aspects of Atlas as they became appropriately mature, while maintaining the expectation that Atlas would become more mature over time. The increments were:

- ***Atlas* 0.25**: The first draft of Atlas based on work done in 2014 was published as Atlas 0.25 in November 2014. It included key elements that explain the effectiveness of systems engineers, and a preliminary explanation of the relationships between those elements. The structure and variables of the proficiency model were also included, along with some initial analysis of career paths.

- ***Atlas* 0.5**: Based on subsequent work done in 2015, Atlas 0.5 was published in December 2015. It reflected further understanding of the elements of Atlas and their inter-relationships. Significant new work was done in the area of career paths and 0.5 incorporated initial efforts to use Atlas to assess the level of proficiency of systems engineers. Atlas 0.5 was mature enough for an individual or an organization to use and gain valuable insights with some guidance from the Helix team.

- ***Atlas* 0.6**: Was an incremental improvement to Atlas 0.5. It contained additional detail and analysis for areas that were less mature in 0.5, namely: mentoring, personal initiatives, and organizational initiatives. Atlas 0.6 was not created as a stand-alone document, but rather as a supplement to 0.5.

- ***Atlas* 1.0**: Atlas 1.0 included a more complete description of the elements of Atlas and their inter-relationships. Atlas 1.0 is believed to be mature enough for independent deployment and assessment by individuals and organizations with little or no guidance from the Helix team. In addition, the frameworks presented in Atlas 1.0 have been validated using data from outside the US DoD, and therefore is believed to be applicable to systems engineers in a variety of domains. This is intentional. Though the initial impetus for the work was based on the needs of the US DoD, the Helix team believes that a more generic framework which benefits all systems engineers, regardless of domain, is both more beneficial to the community at large and, ultimately will benefit the US DoD by setting consistent expectations for practitioners across domains.

- ***Atlas* 1.1**: This is an incremental update to *Atlas* that reflects the teams' learning in 2017.

Atlas 0.25 and Atlas 0.5 were mature enough for trial. The Helix team is aware of five organizations that have used some aspects of Atlas, primarily to assess the proficiency levels and understand the career paths of individual systems engineers within the organization. Feedback and observations from these early use exercises influenced the development of Atlas 1.0 as published here. A glimpse into potential benefits of Atlas deployment, based on trials conducted in 2015 and early 2016, are included throughout this report, with findings related to each element of Atlas reflected in corresponding sections on that element.

1.4 About This Document

This document reflects Atlas 1.1: The Theory of What Makes Systems Engineers Effective. This document:

- Provides an overview of *Atlas 1.1* (Section 2);
- Provides details on the elements of *Atlas 1.1* (Sections 3-8); and
- Provides insights on how these elements come together, specifically referencing the companion documents (Section 9).

As noted above, the *Atlas 1.1 Implementation Guide* and *Atlas Career Path Guidebook* are now companion documents to support individuals and organizations using *Atlas*. With these materials, the Helix team believes that any individual or organization can begin utilizing *Atlas* without direct support from the Helix team. However, the team would be glad to receive feedback and to address any issues, concerns, or questions from the community and can be contacted at helix@stevens.edu.

2 Methodology

The Helix methodology has been extensively documented in previous technical reports (Pyster et al. 2013, Pyster et al. 2014, Pyster et al. 2015, Hutchison et al. 2016). With the goal of keeping *Atlas* 1.1 a streamlined document, the team has chosen not repeat this information here. A full detailing of the methodology can be found in the companion *2017 Helix Technical Report* (SERC-2018-TR-101). In broad strokes:

- The Helix team has spent five years conducting detailed interviews with systems engineers, their peers, and organizational leaders to understand what makes systems engineers effective.

- The research began using a mixed methods approach – the team had expectations on what would make systems engineers effective but did not conduct analysis in a way that looked specifically for those things. Instead, grounded theory was used to "let the data speak" and the patterns in the data became the basis for everything presented here.

- Qualitative analysis methods were used to extract meaning and patterns from the dataset.

Below the team provides a brief description of the dataset on which the findings presented in *Atlas* are based.

2.1 Helix Data Set Updates

From June 2013, when Helix conducted its first site visit for data collection, until November 2017, a total of 335 participants were interviewed from 26 organizations. A brief overview is provided here. For additional detail on the demographics of the sample, please see the *Atlas Career Path Guidebook*.

Figure 2 illustrates the comparison of seniority levels among Helix participants. Prior to 2017, the demographics were the following: junior (19%), mid-level (15%) and senior (66%). Once the information of new participants was included, the percentage of junior systems engineers decreased to (17%). Mid-level increased 2% to a total of 17%. There was no observed change with respect to the percentage of senior systems engineers, which held steady at 66%.

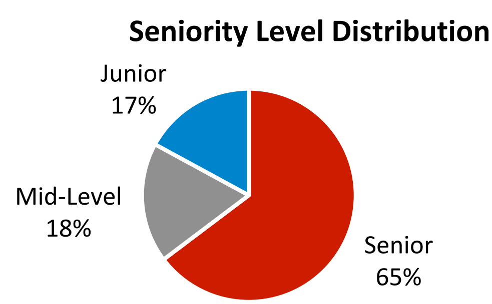

Figure 2. Comparison of seniority level distribution

Helix team has collaborated with 23 organizations. Figure 3 provides an overview of the types of organizations that have participated in Helix. Federally Funded Research and Development Centers (FFRDCs) were added in 2017 as was an additional government organization.

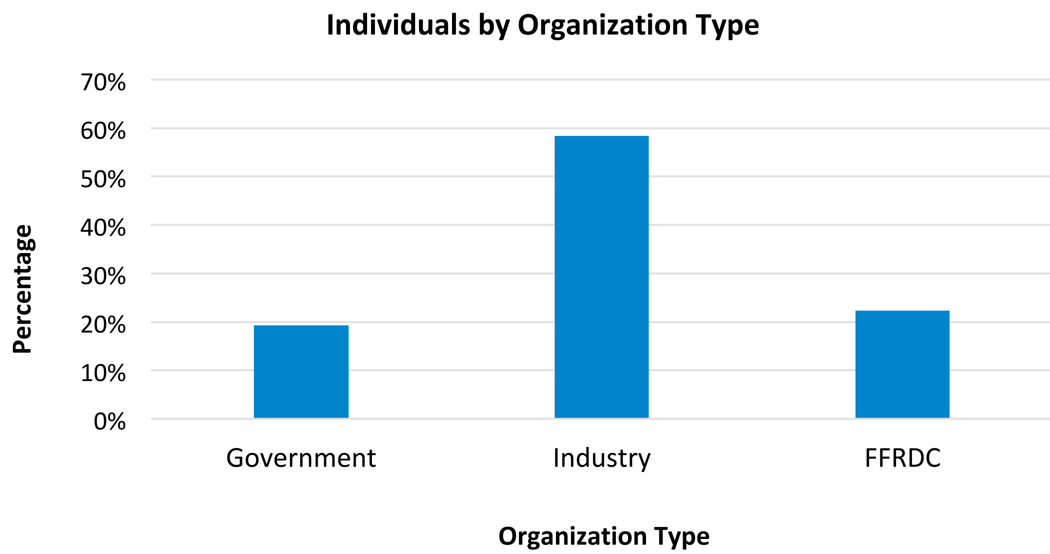

Figure 3. Comparison of organization type distribution

For additional details, please see the dataset analyses in the *Guidebook*.

3 ATLAS 1.1: OVERVIEW

This is unchanged from Atlas *1.1, with the exception of the update to the primary* Atlas *graphic shown in Figure 4.*

Atlas is a set of general principles and ideas that relates to the subject of what makes systems engineers effective and why. In doing so, *Atlas* also provides insights into how individuals can develop into effective systems engineers throughout their careers and what organizations can do to support this development.

3.1 ATLAS OVERVIEW

The overview of *Atlas* in the context of an individual systems engineer employed in an organization is captured in the systemigram illustrated in Figure 4. A systemigram consists of nodes that contain noun phrases, links that contain verb phrases, and is to be read as sentences along the direction of the arrows. The primary sentence is read from the top left node to the bottom right node and presents the main theme of the systemigram. In the ensuing discussions, sentences to be read in the systemigram are italicized, where nodes are represented in square brackets.

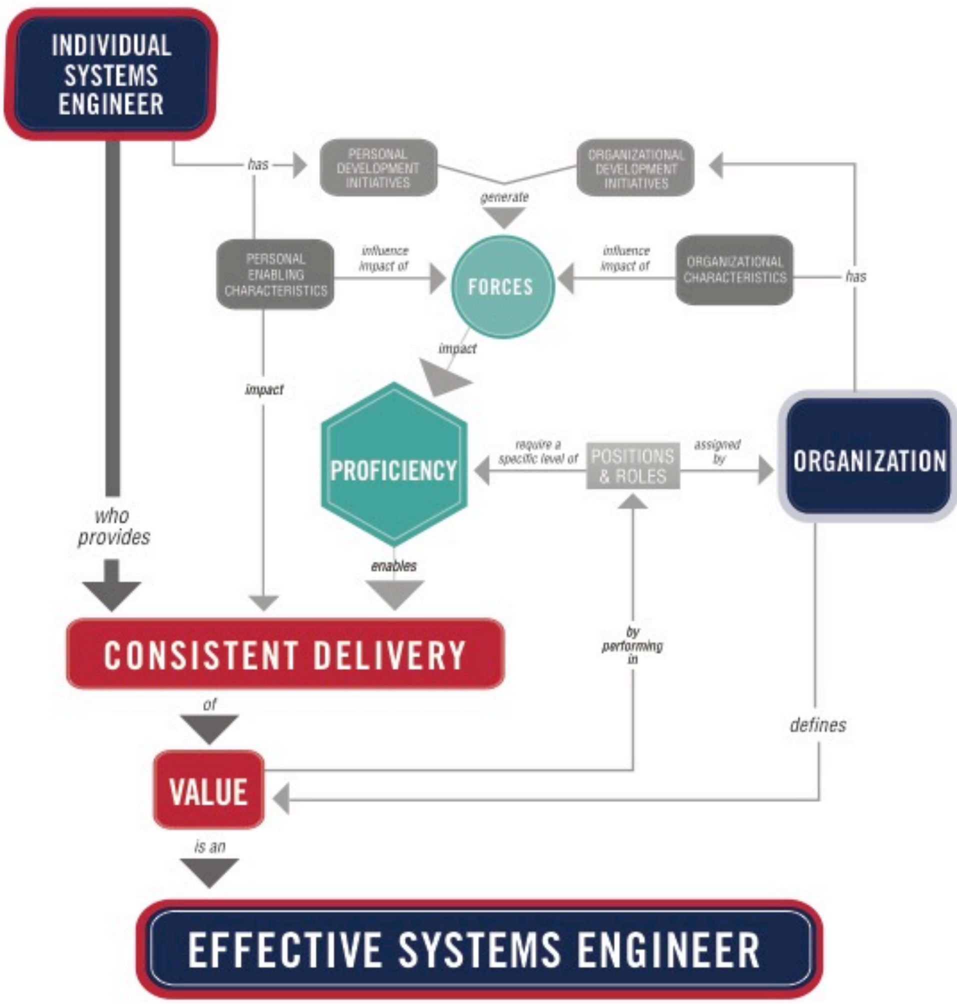

Figure 4. *Atlas 1.1*

From Figure 4 above, it can be seen that the main theme of *Atlas* is: '*[Individual Systems Engineer] who provides [Consistent Delivery] of [Value] is an [Effective Systems Engineer]*'. This fundamental definition of an effective systems engineer hinges on *[Value]*, and it can be seen that '*[Organization] defines [Value]*'. Therefore, it is on the organization to define the value that the systems engineer is expected to provide. Further, the individual systems engineer provides '*[Value] by performing in [Positions and Roles] assigned by [Organization]*'. Therefore, it is again on the organization to establish the position of the systems engineer in terms of roles and responsibilities, keeping in mind that '*[Positions and Roles] require a specific level of [Proficiency] that enables [Consistent Delivery] of [Value]*'.

The core of *Atlas* is the proficiency of the individual systems engineer – what proficiency means, and how it can be improved. '*[Individual Systems Engineer] has [Personal Development Initiatives]*' and '*[Organization] has [Organizational Development Initiatives]*'; together, they '*generate [Forces] that impact [Proficiency]*'. At the same time, '*[Individual Systems Engineer] has [Personal Characteristics] that influence the impact of [Forces]' and '[Organization] has [Organizational Characteristics] that influence the impact of [Forces]'* – these forces may have a positive or a negative influence. Further, both personal enabling characteristics and organizational characteristics '*impact [Consistent Delivery] of [Value]*'; again, the impact can be positive or negative. Amidst all these influences and impacts, the challenge for the individual systems engineer and the organization is to improve the '*[Proficiency] that enables [Consistent Delivery] of [Value]*' to the organization.

The color-coding of the in Figure 4 is designed to show the relationships between various elements of *Atlas* as follows:

- The primary definition for effective systems engineers is highlighted in red.
- Primary actors are in dark blue (individual systems engineer and organization, leading to the desired end state of effective systems engineer).
- Elements related to the skills of systems engineers – the specific skills themselves or how they are developed – are in teal.
- Characteristics of the primary actors are in grey. This includes characteristics of individuals and organizations, the roles and positions of systems engineers defined by organizations, and the initiatives taken by individuals and organizations to improve a effectiveness.

3.2 Dynamic Aspect of *Atlas*

The *Atlas* overview illustrated in Figure 4 can be considered as a quasi-static snapshot in time, but many of the elements of *Atlas* are dynamic in nature. The level of proficiency of an individual systems engineer is not fixed, but is constantly changing due to the impact of forces over time. Similarly, other elements of *Atlas*, including characteristics and initiatives of the individual systems engineer and of the organization, continue to change over time. Further, as the level of proficiency of an individual systems engineer increases over time, the organization is likely to place that systems engineer into different positions.

This dynamic aspect of *Atlas* is not captured in the overview, but is reflected in the career paths of individuals over time, where an individual's career path is the precise combination of the forces they undergo in the positions and roles they perform in over their entire career.

Leading up to the publication of *Atlas 1.0*, the Helix team defined methods to depict and analyze the career paths of systems engineers and used those methods to analyze the systems engineers in its interview sample, and how those systems engineers are shaped by the impact of forces and positions

and roles over time. Notionally, this is reflected in Figure 5.

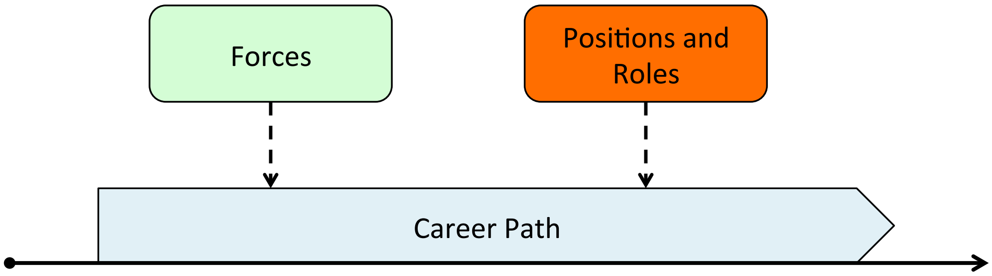

Figure 5. Career Path: A Dynamic View of *Atlas*

The Helix team has defined methods to depict and analyze the career paths of systems engineers. The team used those methods to analyze the systems engineers in its interview sample and to understand how those systems engineers are shaped by the impact of forces and positions & roles over time. These are reflected in the companion *Atlas Career Path Guidebook*.

4 SYSTEMS ENGINEERS PROVIDE CRITICAL VALUES

The discussion of the values has matured to improve clarity. In addition, the activity of functional architecture has moved from "Requirements Engineer" to "Systems Architect".

The broad question that Helix is trying to address is: 'How can an organization develop effective systems engineers?' The key term in this question, in addition to a consistent understanding of who is a systems engineer, is '**effective**'. When initially asked who an 'effective systems engineer' was, interviewees tended to give the response '*one who develops (or supports development of) systems within time, cost, and schedule constraints*'. This definition was not very insightful, and hence Helix developed an alternative definition – an effective systems engineer is '*someone who consistently delivers value by performing systems engineering activities*'. This definition introduced the term 'value', and thus provided a context for effectiveness. Of course, value by itself is a subjective term, and was not something that Helix wanted to define up front. Instead, Helix wanted to understand what systems engineers said was the value they provided and to understand what non-systems engineers said was the value that systems engineers provided.

The Helix team probed on the concept of value in 100% of the interviews conducted. The discussion of value took two general forms: an individual's perspective of the primary value that she provides as a systems engineer and an individual's perspective of the overarching value that systems engineers in her organization provide. Some individuals answered the value question in ways more readily linked with *proficiency* than value; for example, they might have referenced communication skills or deep understanding of systems engineering processes. As indicated above, a number of systems engineers also defined value in terms of overall project success ("on time, within budget"), which does not allow specific insights for systems engineers versus project managers or any other personnel who support the project. After filtering these types of responses, there were 313 individual excerpts on the value that systems engineers provide offered from 85 individual systems engineers.

effectiveness – the ability to consistently deliver value.
systems engineer – an individual who performs systems engineering activities and is recognized (either formally or informally) by his or her organization for her ability to perform these activities.
effective systems engineer – someone who consistently delivers value by performing systems engineering activities in positions assigned by the organization.
value – the benefits gained through the application of systems engineering activities, as distinct from benefits gained through other disciplines.

The key values identified are provided in the list below. The main bullets state the overarching values that systems engineers provide; the sub-bullets are the ways these values are achieved, often discussed as enabling or lower-order values. Percentages reflect the percent of the data related to a given value or the relationship between values. So for example, the first value, "Keeping and maintaining the system vision", was described in 11% of the excerpts on value. However, in 39% of the areas where "Keeping and maintaining the system vision" was discussed, understanding of the customer's true requirements was described as a key enabling value. In some instances, percentages are not provided; these areas require additional analysis.

The primary values that systems engineers provide – as consistently stated across organizational and domain lines – include:

- **Keep and maintain the system vision.** Get the true requirements from the customer, see

relationships between the disciplines, help team members understand those relationships, and provide the big picture perspective for the system. This involves understanding the system vision and explaining it well to the team in a way in which each team member understands their contribution to realizing the vision.

- **Translate technical jargon into business or operational terms and vice versa.** Translate highly technical information from subject matter experts into common language that other stakeholders can understand, as well as translating operational concepts, customer needs, and customer desires into language that makes sense for both engineers and program managers.
- **Enable diverse teams to successfully develop systems.** Bring together a diverse team of engineers and subject matter experts; understand the strengths of each team member and draw on those strengths; rally the team around the common vision; identify and address areas of concern for team integration.
- **Manage emergence in both the project and the system.** Project into the future, which includes staying "above the noise" of day-to-day development issues; communicate the future well to aid decision making that leverages positive emergence and minimizes negative emergence.
- **Enable good technical decisions at the system level.** Balance technical risks and opportunities with the desired end result; leveraging the system vision and a solid grasp on the customer's needs in the application of strong problem-solving abilities – particularly the ability to focus on root cause rather than proximal cause.
- **Support the business case for the system.** Balancing traditional project management concerns of cost and schedule with technical requirements; understand and communicate well the position of a system within the organization's or customer's portfolio.

These values represent the combined perspective from all systems engineers across all organizations – a cross section of government and industry organizations from multiple domains. These were seen as the consistent values and no major differences were seen between government and industry or across different domains. However, it is worth noting that the means for delivering value was different.

For example, whether in the defense sector or other sectors, systems engineers in government organizations tended to be more focused on providing value by emphasizing standard processes, while commercial organizations tended to focus more on delivering the "right" end results by asking good questions, generating a vision for the system, and providing the big picture perspective. This ***does not*** mean that systems engineers in government organizations value process over the end result of systems development; instead, it means that in an acquisition environment – which was the context for the majority of government systems engineers – following a rigorous process was seen as a primary way to provide the values listed above and help achieve end results. In commercial companies, process was discussed, but not seen as the primary means for providing value. Systems engineers in commercial companies did state that systems engineers provide value by bringing a logical approach to problem solving and, in some organizations, processes were seen as a way to institutionalize these types of approaches, although with varying degrees of success. It is worth noting that systems engineers in commercial organizations in highly regulated industries tended to emphasize process more strongly than their counterparts in less-regulated industries.

In addition to the primary values, there are several sets of enabling values in Table 1. These are activities systems engineers perform that provide value to a project and, when combined, deliver the primary value. They are in some ways the "how" of the primary value's "what". Notice that some of the enabling values appear several times. Note that Value 2, "Translation" does not have an enabling value. This is

because when the team asked how systems engineers delivered this value, they provided critical *proficiencies*, but nothing that rose to enabling values (which generally require a variety of proficiencies to deliver). In Table 1, number in parentheses represent the percent of individuals fro the Helix sample who discussed the relationship between the enabling value and the primary value.

Table 1. Relationships Between Primary and Enabling Values

#	Primary Values	Enabling Values
1	**Keep and maintain the system vision**	• Get the "true" requirements from the customer and creating alignment between the customer and the project team. (39%) • See relationships between the disciplines and help team members understand and respect those relationships. (33%) • Balance technical risks and opportunities with the desired end result. (36%) • Provide the big picture perspective for the system. (44%)
2	**Translate technical jargon into business or operational terms and vice versa**	
3	**Enable diverse teams to successfully develop systems**	• Effectively understand and communicate the system vision to the team, and ensure that the team is aligned with this vision. (38%) • Help the team to understand the big picture perspective and where they fit within the larger picture. (38%) • Identify areas of concern for integration in advance. (13%)
4	**Manage emergence in both the project and the system**	• Project into the future (14%), which includes staying "above the noise" of day-to-day development issues and identifying pitfalls. • Balance technical problem-solving with the big picture perspective. (43%)
5	**Enable good technical decisions at the system level**	• See the vision for the system and communicate that vision clearly to help teams make good technical decisions. (40%) • Provide the big picture perspective, understanding the system holistically and enabling system-level technical decisions versus decisions made at the component or sub-system level. (22%) • Ensure that decisions made will keep the system on the correct technical path using a solid grasp on the customer's needs. (22%) • Be able to bring together a diverse team of engineers and subject matter experts. (26%) • Be able to focus on root versus proximal causes of technical issues. (26%).
6	**Support the business cases for systems**	• Balance traditional project management concerns of cost and schedule with technical requirements. (41%) • Understand the position of a system within the organization or customer's portfolio and communicating this to the team. (59%)

In *Atlas*, effectiveness is defined as the ability to consistently deliver these values over time. All elements of *Atlas* described below are included because in the dataset, they were linked back to the ability to provide these values.

5 Systems Engineers' Positions and Roles

This discussion has matured and the groupings have been updated slightly since Atlas 1.0.

An individual systems engineer fills a *position* (or holds a title) in an organization, and there are many *roles* that the systems engineer is expected to perform in that position. *Atlas* identifies 17 systems engineering roles; typically, a systems engineer performs a combination of these roles while holding a single position. Starting with the 'twelve systems engineering roles' identified by Sheard (1996). The Helix team recombined, renamed, removed, and added roles to reflect the Helix data collected during interviews about the activities systems engineers perform in organizations today. This was socialized with the community through conference papers and presentations, the Helix workshops, and through early adopter activities with several organizations.

role – a set of specific, related systems engineering activities.
position – the particular arrangement of roles and responsibilities for an individual, as defined and assigned by the organization. Often, positions are equivalent to an individual's title.

5.1 Atlas Roles Framework

Tables 2-4 provide the roles of systems engineers and offers an explanation of how each role came to exist in the framework. For example, "System Integrator" is the role that was previously titled "Glue" in (Sheard 1996) and the name change as well as the rationale for the change is captured below. Tables 2-4 also highlight the *roles* framework developed, consisting of three categories:

- ***Roles Focused on the System Being Developed*** – These roles are what may most quickly come to mind when describing a systems engineer. They align closely with the systems engineering lifecycle and the critical activities systems engineers must enable throughout the lifecycle.
- ***Roles Focused on SE Process and Organization*** – These roles focus on the organizational context in which systems engineering works and the critical role of systems engineers in providing guidance on how systems engineering should be used.
- ***Roles Focused on Teams that Build Systems*** – Systems engineering does not occur in a vacuum and is, instead, an intensely social activity. The roles in this category focus on enabling diverse, multi-disciplinary teams to be successful.

The categories help distinguish between the major types of activities that systems engineers provide.

Table 2. Roles Focused on the Systems Being Developed

Role Name	Role Description
Concept Creator	Individual who holistically explores the problem or opportunity space and develops the overarching vision for a system(s) that can address this space. A major gap pointed out to the Helix team – particularly when working to implement the findings of Helix – has been that of the development of an overarching system vision. This is a critical first step in the systems lifecycle, and several organizations stated that they believed it needed to be separately called

Role Name	Role Description
	out. In addition, when looking to the future of what systems engineers need to do (e.g., INCOSE Vision 2025 (2015)), the focus on early engagement and setting the vision was deemed critical.
Requirements Owner	Individual who is responsible for translating customer requirements to system or sub-system requirements. **Note: This is updated from Atlas 1.0. Sheard (1996) also included the activities around functional architecture in this role. However, in working with the community, this has caused some confusion as to the differences between this role and that of "System Architect". The Helix team believes that grouping all architecture activities together will improve clarity on the roles.**
System Architect	Individual who owns or is responsible for the architectures of the system; this including functional and physical architectures. **Note: This is updated from *Atlas* 1.0. This is an update of Sheard's "System Designer" role (1996). There was concern both at community events and during later interviews that nowhere in the presented framework did the critical role of systems engineers in architecture come out clearly. Some also argued that "Design" gave the impression that this role focuses specifically on the details of systems design over architecture.**
System Integrator	Individual who provides a holistic perspective of the system; this may be the 'technical conscience' or 'seeker of issues that fall in the cracks' – particularly, someone who is concerned with interfaces. Likewise, there was concern over the word "Glue", which many expressed was not clearly descriptive enough.
System Analyst	Individual who provides modeling or analysis support to system development activities, and helps to ensure that the system as designed meets he specification. This is unchanged from Sheard's roles (1996).
Detailed Designer	Individual who provides technical designs that match the system architecture; an individual contributor in any engineering discipline who provides part of the design for the overall system. This is an addition based on the Helix data. While systems engineers do not always get involved with detailed design, in smaller organizations or on smaller projects it is more common. Likewise, systems engineers who had played this role explained that it was critical in developing their own technical and domain expertise as well as in understanding the design approaches of classic engineers.
V&V Engineer	Individual who plans, conducts, or oversees verification and validation activities such as testing, demonstration, and simulation. This is unchanged from Sheard's roles (1996).

Role Name	Role Description
Support Engineer	Individual who performs the 'back end' of the systems lifecycle, who may operate the system, provide support during operation, provide guidance on maintenance, or help with disposal. This was previously titled "Logistics and Operations Engineer" in Sheard (1996). However, in interviews and at community events, the Helix team received feedback that using this title gave the impression that this role was limited and did not encompass the full spectrum of systems engineers' activities at system deployment or post-deployment. Likewise, in several organizations, "logistics" and "operations" were seen as separate disciplines from systems engineering, which caused some contention in discussions. The renaming of this category is intended to address these issues.

Table 3. Roles Focused on Process and Organization

Role Name	Role Description
Systems Engineering Champion	Individual who promotes the value of systems engineering to individuals outside of the SE community – to project managers, other engineers, or management. This may happen at the strategic level or could involve looking for areas where systems activities can provide a direct or immediate benefit on existing projects. Sheard recommended that a role such as this, labeled in her work as "Systems Engineering Evangelist", be added in (2000).
Process Engineer	Individual who defines and maintains the systems engineering processes as a whole and who also likely has direct ties into the business. This individual provides critical guidance on how systems engineering should be conducted within an organization context. This is unchanged from Sheard's roles (1996).

Table 4. Roles Focused on the Teams That Build Systems

Role Name	Role Description
Customer Interface	Individual who coordinates with the customer, particularly for ensuring that the customer understands critical technical detail and that a customer's desires are, in turn, communicated to the technical team. This is unchanged from Sheard's roles (1996).
Technical Manager	Individual who controls cost, schedule, and resources for the *technical* aspects of a system; often someone who works in coordination with an overall project or program manager. This is unchanged from Sheard's roles (1996).
Information Manager	Individual who is responsible for the flow of information during system development activities. This includes the systems management activities of configuration management, data management, or metrics. This is unchanged from Sheard's roles (1996).

Role Name	Role Description
Coordinator	Individual who brings together and brings to agreement a broad set of individuals or groups who help to resolve systems related issues. This is a critical aspect of the management of teams. This is unchanged from Sheard's roles (1996).
Instructor/Teacher	Individual who provides or oversees critical instruction on the systems engineering discipline, practices, processes, etc. This can include the development or delivery of training curriculum as well as academic instruction of formal university courses related to systems engineering. While any discipline could conceivably have an instructor role, this denotes a focus on systems and is a critical component in the development of an effective systems engineering workforce. This is an addition to the Sheard roles (1996 and 2000).

The role of "Classified Ad" systems engineer, as defined by Sheard (1996) was dropped from this framework. "Classified Ad" was a placeholder role Sheard used to acknowledge the many job postings for "systems engineers" reflected IT network or computer specialists (e.g., network systems engineer, IT systems engineer, or Microsoft systems engineer). In the Helix sample, none of the systems engineers for whom roles data was collected played this role, either currently or in the past. In addition, when this role was presented at various community events (Helix workshops in 2014, 2015, and 2016; presentations on Helix at INCOSE (Lipizzi, 2015, Jauregui, 2016), there was a strong recommendation to remove it from the framework to highlight what systems engineers do and to draw a clear distinction from positions that may be titled "systems engineer" but which do not bear resemblance to the practice of systems engineering.

Tables 2-4 outline the *systems engineering* roles. However, there were a few roles that were commonly seen throughout the Helix data sample. These are roles that may frequently be played by systems engineers. These include:

- *Organizational/Functional Manager* - Individual who is responsible for the personnel management of systems engineers or other technical personnel in a *business* – not a project or program – setting.
- *Program/Project Manager* - Individual who is not *directly* responsible for the technical content of a program, but works closely with technical experts and other systems engineers while maintaining overall project cost and schedule.

These roles, while not systems engineering roles, are things that many systems engineers do throughout their careers and which may help systems engineers develop some critical skills. Figure 6 provides a simple Venn diagram showing, from the Helix data, the overlap between systems engineering roles and roles held by systems engineers.

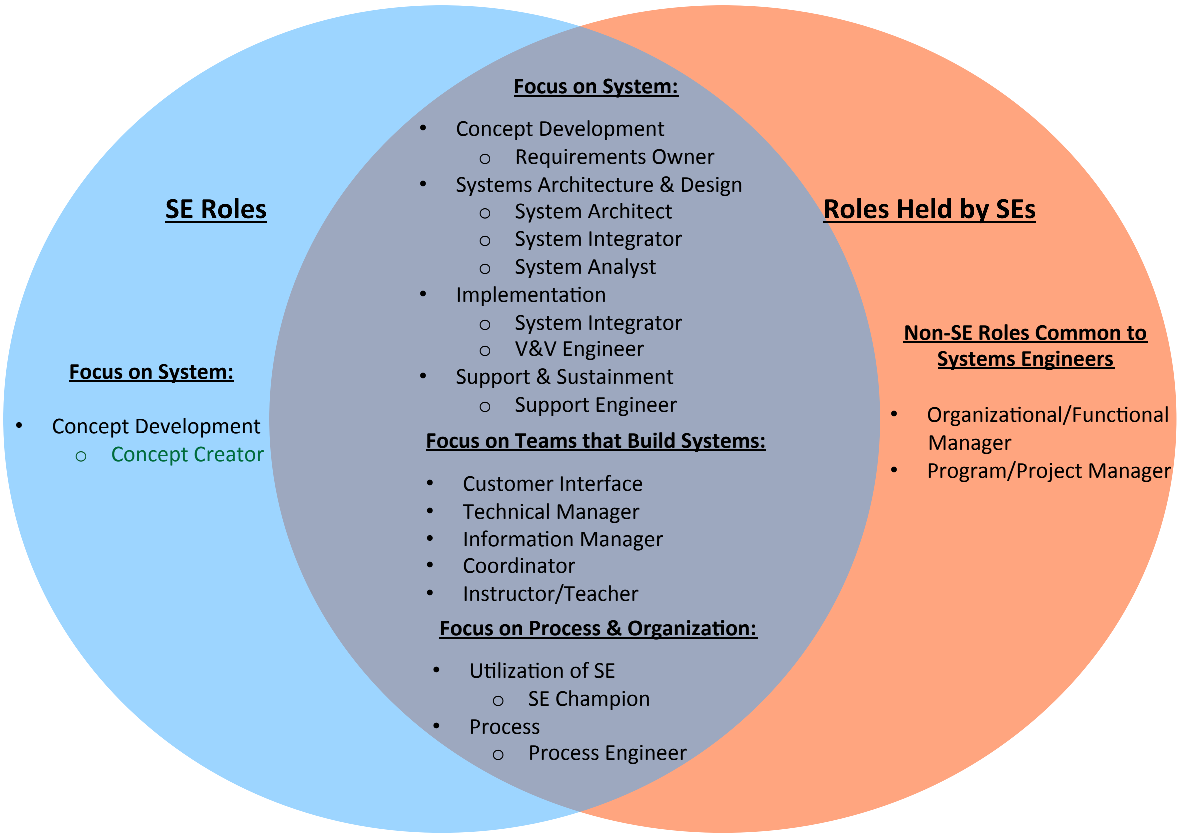

Figure 6.The overlap between SE roles and Roles Held by Systems Engineers in the Helix Sample

It may be surprising that one of the SE roles, "Concept Creator" (shown in green in Figure 6), is not a role that systems engineers in the Helix sample commonly played. A small number of individuals in the Helix sample did play these roles, but not enough, initially, to add this to the framework. The addition of this role was based on community feedback and work on implementation with several organizations. The Helix team believes that the primary reason that "Concept Creator" did not come out strongly in the sample is due to the organizations in which they work. In each of the government organizations that participated, systems engineers have been part of the acquisition workforce. When asked if they participated in initial concept definition, most explained that this was done before they were assigned to the system. Systems engineers at many industry organizations, particularly those within the DIB, expressed a similar view – that this early vision-setting happened before systems engineers got involved.

In looking to the future of systems engineers, there is a push for them to be included more in concept design. Clearly, concept development work is part of systems engineering as it is critical for successful systems, and one would assume that this would be an important role for systems engineers. This is reflected in strategic documents such as INCOSE *Vision 2025* (2014) as well as in the goals and desires of several organizations working to implement Helix findings and individual systems engineers. This is the rationale for inclusion in the *Atlas* roles framework.

5 The Proficiencies of Systems Engineers

The proficiency framework presented here has been matured in several small ways based on the interest and feedback of the community and lessons learned as organizations and individuals worked to apply Atlas. When presented at community events, there were commonly questions about the overlap between Proficiencies and Personal Characteristics. Proficiencies are knowledge, skills, abilities, behaviors, and cognitions that an individual utilizes to perform systems engineering. There are clear ways to grow proficiencies and individuals can be guided on growth paths for these. Personal characteristics, however, are more internally focused and much more difficult to grow. This does not mean that an individual can not improve. Take, for example, the personal characteristic of "self-awareness". An individual can be told that self awareness is important, given tools to improve self-awareness, and participate in 360° feedback to give them information to improve their self-awareness – but they may or may not become more self aware. While this is true to some extent for anything, the Helix team views proficiencies as skills that are more easily influenced externally versus personal characteristics, which are largely dependent on internal factors.

> ***proficiency*** – the quality or state of knowledge, skills, abilities, behaviors, and cognition.
> ***proficiency area*** - grouping of related knowledge, skills, abilities, behaviors, and/or cognition.
> ***proficiency level*** –extent to which an individual has attained the knowledge, has the skill and ability to perform a task, or has demonstrated relevant behaviors, or cognitions.

The proficiency model of *Atlas*, captures the knowledge, skills, abilities, behaviors, patterns of thinking, and abilities that are critical to the effectiveness of systems engineers.

- **Proficiency** is the quality or state of knowledge, skills, abilities, behaviors, and cognition.
- Proficiency **Areas** are groupings of related knowledge, skills, abilities, behaviors, and/or cognition.
  - Each Proficiency Area is comprised of **Categories**, which are specific types of knowledge, skills, abilities, behaviors, and cognition with shared characteristics.
    - *Some* categories are further refined into **Topics**, which are the most discrete areas of proficiency included in *Atlas*.
- For each proficiency area, there are **Levels**, which describe the extent to which an individual has attained certain knowledge, has the ability to perform a certain skill, or has demonstrated relevant abilities, behaviors, or cognition. Loosely, a scale of 0 to 5 is used to indicate the level of proficiency at the area level, where 5 indicates the highest possible or "Expert" proficiency.

The *Atlas* proficiency model, along with identified proficiency levels, enables a proficiency profile to be created for an individual at any point in time, as illustrated in Figures 7, 8, and 9, below. Currently, proficiency levels are documented only for proficiency **Areas**.

5.1 *Atlas* Proficiency Model

The *Atlas* proficiency model consists of six proficiency areas based on the Helix interview data, as shown in Figure 7 below.

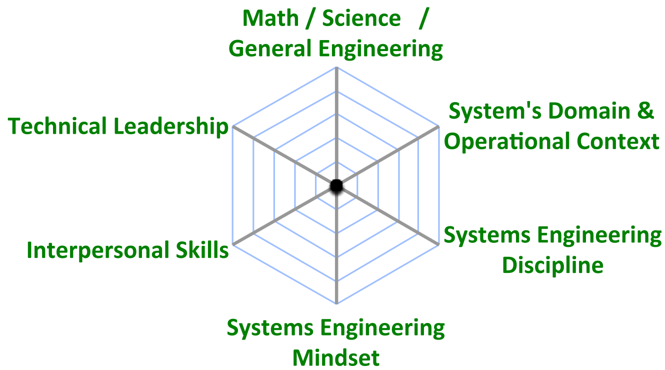

Figure 7. Proficiency Areas for Systems Engineers

1. ***Math/Science/General Engineering:*** Foundational concepts from mathematics, physical sciences, and general engineering;
2. ***System's Domain & Operational Context:*** Relevant domains, disciplines, and technologies for a given system and its operation;
3. ***Systems Engineering Discipline***: Foundation of systems science and systems engineering knowledge;
4. ***Systems Engineering Mindset***: Skills, behaviors, and cognition associated with being a systems engineer;
5. ***Interpersonal Skills***: Skills and behaviors associated with the ability to work effectively in a team environment and to coordinate across the problem domain and solution domain; and
6. ***Technical Leadership***: Skills and behaviors associated with the ability to guide a diverse team of experts toward a specific technical goal.

Proficiency areas 1 to 3 consist of primarily 'hard' or technically based skills, while proficiency areas 4 to 6 consist primarily of the 'soft' or interdisciplinary skills. The six proficiency areas in *Atlas* are further divided into categories and, in some cases, into topics, as shown in Table 5. Each of the proficiency areas is elaborated in the subsequent sections.

Table 5. Atlas Proficiency Areas, Categories, and Topics

Area	Category	Topic
1. Math / Science / General Engineering	**1.1.** Natural Science Foundations	
	1.2. Engineering Fundamentals	
	1.3. Probability and Statistics	
	1.4. Calculus and Analytical Geometry	
	1.5. Computing Fundamentals	

Area	Category	Topic
2. Systems' Domain & Operational Context	2.1. Principal and Relevant Systems	< List of Principal and Relevant Systems >
	2.2. Familiarity with Principal System's Concept of Operations (ConOps)	
	2.3. Relevant Domains	< List of relevant Domains >
	2.4. Relevant Technologies	< List of relevant Technologies >
	2.5. Relevant Disciplines and Specialties	< List of relevant Disciplines and Specialties >
	2.6. System Characteristics	< List of applicable System Types, Scales, and Levels >
3. Systems Engineering Discipline	3.1. Lifecycle	3.1.1 Lifecycle Models 3.1.2 Concept Definition 3.1.3 System Definition 3.1.4 System Realization 3.1.5 System Deployment and Use 3.1.6 Product and Service Life Management
	3.2. Systems Engineering Management	3.2.1 Planning 3.2.2 Risk Management 3.2.3 Configuration Management 3.2.4 Assessment and Control 3.2.5 Quality Management
	3.3. SE Methods, Processes, and Tools	3.3.1 Balance and Optimization 3.3.2 Modeling and Simulation 3.3.3 Development Process 3.3.4 Systems Engineering Tools
	3.4. Systems Engineering Trends	3.4.1 Complexity 3.4.2 Model Oriented Systems Engineering 3.4.3 Systems Engineering Analytics 3.4.4 Agile Systems Engineering
4. Systems Engineering Mindset	4.1. Big-Picture Thinking	
	4.2. Paradoxical Mindset	4.2.1 Big-Picture Thinking and Attention to Detail 4.2.2 Strategic and Tactical 4.2.3 Analytic and Synthetic 4.2.4 Courageous and Humble 4.2.5 Methodical and Creative
	4.3. Adaptability	
	4.4. Abstraction	
	4.5. Foresight and Vision	
5. Interpersonal Skills	5.1. Communication	5.1.1 Audience 5.1.2 Content 5.1.3 Mode
	5.2. Listening and Comprehension	

Area	Category	Topic
	5.3. Working in a Team	
	5.4. Influence, Persuasion, and Negotiation	
	5.5. Building a Social Network	
6. Technical Leadership	6.1. Building and Orchestrating a Diverse Team	
	6.2. Balanced Decision Making & Rational Risk Taking	
	6.3. Guiding Diverse Stakeholders	
	6.4. Conflict Resolution & Barrier Breaking	
	6.5. Business and Project Management Skills	
	6.6. Establishing Technical Strategies	
	6.7. Enabling Broad Portfolio-Level Outcomes	

5.1.1 Area 1: Math/Science/General Engineering

A good understanding of math, science, and general engineering is a critical foundation for effective systems engineers; but this understanding is largely 'assumed' in a systems engineer when joining the workforce. However, it is on this foundation that further understanding of the categories under Proficiency Area 2: *Systems' Domain & Operational Context* is built.

The *Graduate Reference Curriculum for Systems Engineering (GRCSE®)* defines the types of prerequisite knowledge individuals should have before entering a master's program in systems engineering (Pyster et al. 2015). Since limited insight was obtained from Helix data collection and analysis for this proficiency area, GRCSE is used to identify and define the categories in this area:

1.1. **Natural Science Foundations:** Basic concepts and principles of one of the natural science disciplines (e.g., physics, biology, chemistry, etc.); includes laboratory work that involves experimental techniques, the application of the scientific method, and comprehension of appropriate methods for data quality assurance and analysis.

1.2. **Engineering Fundamentals:** The nature of engineering, branches of engineering, the design process, analysis and modeling, the role of empirical and statistical techniques, problem solving strategies, and the value of standards; some level of practical experience is expected, whether through capstones, internships, or course projects. Practical experience should include the application of engineering fundamentals in a specific domain context.

1.3. **Probability and Statistics:** Basic probability theory, random variables and probability distributions, estimation theory, hypothesis testing, regression analysis, and analysis of variance.

1.4. **Calculus and Analytical Geometry:** Theory and application of differential and integral calculus methods and operations; study of techniques for describing, representing, and analyzing geometric objects (coordinate systems, algebraic models, graphing).

1.5. **Computing Fundamentals**: Overview of computer organization (computer architecture, operating systems, and programming languages), algorithms, and data structures; software engineering fundamentals (lifecycle models, quality, cost, and schedule issues); and development of a software unit (design, coding, and testing).

Proficiencies in Area 1: Math/Science/General Engineering may be considered as the general foundation that is provided in any undergraduate engineering degree. Advanced levels of these topics are included in the topics of Area 2, in the context of the system of concern. For an individual without a formal undergraduate degree in engineering, obtaining the proficiencies in Area 1 could happen through experience, training, or mentoring.

5.1.2 Area 2: System's Domain & Operational Context

The second proficiency area is *System's Domain & Operational Context*, which contains the relevant domains, technologies, disciplines, specialties, and characteristics for a given system, and the operation of that system. This proficiency area strongly corresponds to the organization and the systems that its systems engineers work on. If an individual transitions to a new system, the proficiency level may change depending on familiarity with the new relevant domains, technologies, and disciplines. The categories for this proficiency area are defined below:

2.1. **Principal and Relevant Systems**: *Principal* systems are those systems that are of primary interest to the organization. High levels of proficiency in those specific systems are desired by the organization. If a combat ship were the principal system, relevant systems could be submarines and aircraft carriers, which are types of combat ships.

2.2. **Familiarity with Principal System's Concept of Operations (ConOps)**: A system's concept of operations (ConOps) of how systems in the domain are used and deliver value, especially those systems on which the individual personally works. Familiarity with the principal system's ConOps is of particular interest, though familiarity with the ConOps of other related systems may also be helpful.

2.3. **Relevant Domains**: *Domain* refers to the overarching area of application of the system; this includes things such as space, aerospace, marine, communication, finance, etc. Proficiency in related domains outside the primary one may enable an individual to be more effective in the primary domain. For example, experience in space systems may enable a systems engineer to work in aerospace systems more readily than an engineer who is proficient primarily in finance systems.

2.4. **Relevant Technologies**: Within the context of a system, there are specific technologies that are relevant. For example, on a marine system, these may be technologies such as gas turbine, radar, and sonar systems; and each technology has its own terminology, challenges, etc.

2.5. **Relevant Disciplines and Specialties**: Disciplines are fundamental areas of education or expertise that are foundational to a system. For example, for a communications system,

electrical engineering will be an important discipline to understand, while civil engineering will be less relevant. Specialties are disciplines that support systems engineering by applying cross-cutting knowledge. Specialties include Reliability, Availability, and Maintainability (RAM), Human Systems Integration, Safety Engineering, Affordability and other related topics.

2.6. System Characteristics: Three characteristics are considered in *Atlas*:

- **System Type**: Types of systems include technical systems, social systems, human systems, physical systems, cyber systems, and any combination of these. Another classification of system types includes product systems, service systems, and enterprise systems.
- **System Scale:** Systems can be anywhere from a nano level to a distributed global or enterprise level. A generic systems engineering development process may be applicable to systems at any scale.
- **System Scope:** What can be seen as a system from one perspective, could be a subsystem from another perspective. The levels of a system could range from component/element, subsystem, system, and platform or system of systems.

5.1.3 AREA 3: SYSTEMS ENGINEERING DISCIPLINE

The third proficiency area is *Systems Engineering Discipline*. The categories below were developed based on data from Helix interviews about critical systems engineering knowledge and skills. The category names are taken from the *Guide to the Systems Engineering Body of Knowledge (SEBoK)* (BKCASE Editorial Board 2015). Some of the categories are further expanded into topics.

3.1. Lifecycle: *The organized collection of activities, relationships and contracts that apply to a system-of-interest during its life* (Pyster 2009). This is a roll up of knowledge about lifecycles and proficiency in specific aspects of the lifecycle. Topics 3.1.2 – 3.1.6 below, represent generic lifecycle phases in system development:

3.1.1. Lifecycle Models: *A framework of processes and activities concerned with the lifecycle that may be organized into stages, which also acts as a common reference for communication and understanding* (ISO/IEC/IEEE 15288). Lifecycle Models include the Vee model; iterative models such as the spiral development model; formal acquisition models (e.g., as defined in DoD 5000.2 2013); or less formal acquisition models (e.g., quick reaction capability or internal research and development (IR&D) models).

3.1.2. Concept Definition: *A set of core technical activities of systems engineering in which the problem space and the needs of the stakeholders are closely examined* (BKCASE Editorial Board 2016). This consists of analysis of the problem space, business or mission analysis, and the definition of stakeholder needs for required services.

3.1.3. System Definition: *A set of core technical activities of systems engineering, including the activities that are completed primarily in the front-end portion of the system design.* (BKCASE Editorial Board 2016) This consists of the definition of system requirements, the design of one or more logical and physical architectures, and analysis and selection between possible solution options.

3.1.4. System Realization: *The activities required to build a system, integrate disparate system*

elements, and ensure that a system both meets the needs of stakeholders and aligns with the requirements identified in the system definition stage (BKCASE Editorial Board 2016). This includes implementation as well as integration, verification, and validation (IV&V).

3.1.5. System Deployment and Use: *A set of core technical activities of systems engineering to ensure that the developed system is operationally acceptable and that the responsibility for the effective, efficient, and safe operations of the system is transferred to the owner* (BKCASE Editorial Board 2016). Considerations for deployment and use must be included throughout the system lifecycle. Activities within this phase include deployment, operation, maintenance, and logistics.

3.1.6. Product and Service Life Management: Deals with the overall lifecycle planning and support of a system (BKCASE Editorial Board 2016). The life of a product or service often spans a considerably longer period of time than what is required to design and develop the system. This stage includes service life extension, updates, upgrades, and modernization, and disposal and retirement.

3.2. Systems Engineering Management: *Managing the resources and assets allocated to perform systems engineering, often in the context of a project or a service, but sometimes in the context of a less well-defined activity. Systems engineering management is distinguished from general project management by its focus on the technical or engineering aspects of a project* (BKCASE Editorial Board 2016). The topics contained in the *Systems Engineering Management* category are defined below:

3.2.1. Planning: Planning involves developing and integrating technical plans to achieve the technical project objectives within the resource constraints and risk thresholds. This involves the success-critical stakeholders to ensure that necessary tasks are defined with the right timing in the lifecycle in order to manage acceptable risks levels, meet schedules, and avoid costly omissions (BKCASE Editorial Board 2016).

3.2.2. Risk Management: Organized, analytic process to identify what might cause harm or loss (identify risks); to assess and quantify the identified risks; and to develop and, if needed, implement an appropriate approach to prevent or handle causes of risk that could result in significant harm or loss (ISO/IEC/IEEE 24765:2010 – SEVocab).

Note: In presenting this framework at community events, the Helix team has received ample feedback that Risk Management is seen as a critical aspect of systems engineering and there was concern that placing it as a topic under systems engineering management may somehow negate its importance. The team reviewed this and believes that its placement as a topic is still appropriate, as it is one topic within the overarching category of "Systems Engineering Management". Instead, the team recommends that an organization that wishes to emphasize the importance of risk management within its context should do so through tailoring.

3.2.3. Configuration Management: A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements (ISO/IEC/IEEE 24765:2010 – SEVocab).

3.2.4. Assessment and Control: This process involves determining and initiating the appropriate handling strategies and actions for findings and/or discrepancies that are

uncovered in the enterprise, infrastructure, or lifecycle activities associated with the project (BKCASE Editorial Board 2016).

3.2.5. Quality Management: Whether a systems engineer delivers a product, a service, or an enterprise, the deliverable should meet the needs of the customer and be fit for use. Such a deliverable is said to be of high quality. The process to assure high quality is called quality management (BKCASE Editorial Board 2016).

3.3. SE Methods, Processes, and Tools: *A systems engineering method is set of activities, methods, practices, and transformations that people use to develop and maintain systems and associated products* (SEI 2007). Processes generally refer to the specific guidelines an organization develops for implementing systems engineering methods; tools refer to software programs that are designed to support systems engineering activities. The topics contained in the *SE Methods, Processes, and Tools* category are outlined below:

3.3.1. Balance and Optimization: Specialty engineers often focus on the details and optimization of their specific components of the system, but that optimization of individual components often leads to a less-than-optimal system solution. Systems engineers, therefore, have to be able to balance the desire for component optimization with the optimization for the system overall, which often requires sub-optimization for one or more components.

3.3.2. Modeling and Simulation: *A model is a simplified representation of a system at some particular point in time or space intended to promote understanding of the real system. A simulation is the manipulation of a model in such a way that it operates on time or space to compress it, thus enabling one to perceive the interactions that would not otherwise be apparent because of their separation in time or space* (Bellinger 2004). This topic represents and individual's ability to understand and perform modeling and simulation; this understanding is more fundamental than the ability to use software tools that support modeling and simulation.

3.3.3. Development Processes: Each organization has its own processes that govern the development of systems. It is important for systems engineers to understand generic systems engineering processes, but also the specific processes being used for development within the organization or domain.

3.3.4. Systems Engineering Tools: Systems engineers need to be able to utilize tools to support overall system development and to perform the systems engineering development process. Tools may include requirements management and other tools that assist with project life management (PLM).

3.4. Systems Engineering Trends: *Current and future trends in performing Systems Engineering, that modify the way systems are developed.*

3.4.1. Complexity: Complexity of a system is generally understood to exist not in a higher order scale or level of a system, but rather in the higher order of interactions between system elements, disciplines, or technologies, and the properties that emerge out of these interactions that are not present in the individual elements. One categorization of complexity includes structural complexity, dynamic complexity, and socio-political complexity; while another identifies two kinds of complexity: disorganized complexity and organized complexity (SEBoK authors, "Complexity", 2016).

3.4.2. Model Oriented Systems Engineering: Model Based Systems Engineering (MBSE) is a

theme that is being increasingly adopted in systems engineering, where models are used to describe various elements of systems and the systems development process. Model Oriented Systems Engineering (MOSE) goes beyond MBSE, and presents a holistic model-based approach that integrates operational, technical, programmatic and business dimensions as well.

3.4.3. **Systems Engineering Analytics:** The increasing ability to collect, store, analyze, and gain insights from large quantities of data has significantly improved the area of analytics in general. This perspective can also be applied to systems engineering, where complex phenomena within systems and systems development can be measured and analyzed.

3.4.4. **Agile Systems Engineering:** The shrinking of systems engineering development lifecycles, increasingly uncertain and rapidly changing requirements and operational environments of modern systems, has led to the development and adoption of agile systems engineering approaches.

5.1.4 Area 4: Systems Mindset

The fourth proficiency area is *Systems Mindset*, which is primarily focused on patterns of thinking, perceiving, and approaching a task that are particularly relevant to systems engineers. The title for this has changed since *Atlas* 1.0 (previously "Systems Engineering Mindset"). At several events where *Atlas* was presented to the community, individuals commented that the previous proficiency areas covered the "engineering" mindset and that this area really focuses on holism and integration, which was more of a systems view than strictly a systems engineering one. The team agreed. This change better reflects the content of the proficiency area.

The categories included in this area are:

4.1. **Big-Picture Thinking**: Also referred to as 'systems thinking' and 'holistic thinking', this includes the ability to step back and take a broader view of the problem at hand; this is an important and essential characteristic of systems engineers. 'Big-picture' could refer to a broader perspective along many different dimensions: the system as a whole including interfaces and integration, and not limited to any sub-system or component; the system while in operation, and its interactions with other systems and the operating environment; the entire lifecycle of the system, and not limited to the current stage of the system; the development program in the context of the organization and all its other development programs; the end goal or solution to the problem at hand; the perspectives of different stakeholders; and the technical as well as business perspectives. A systems engineer is usually *the* person to bring this broader perspective, while classic engineers and subject matter experts often tend to be narrowly focused on their area of interest. Systems engineers are not only called to provide this big-picture perspective themselves, but to also enable others to see this bigger picture.

4.2. **Paradoxical Mindset**: *The ability to hold and balance seemingly opposed views, and being able to move from one perspective to another appropriately*. Typically, an engineer may hold one view or the other, but rarely *both*. By having this paradoxical mindset, a systems engineer contributes value that is not usually expected from others. The opposing-concept pairs are:

4.2.1. **Big-Picture Thinking *and* Attention to Detail:** Big-picture thinking provides the broader higher-level perspective; at the same time, a systems engineer is also required to pay attention to the details of how things work and how they come together in a system.

4.2.2. **Strategic *and* Tactical:** Systems engineers need to be strategic, focused on the end result of 'vision' for the system, but also need to handle the tactical day-to-day activities and decisions required to reach that vision. They must also be able to appreciate "how what is done today is going to affect things downstream". A related concept pair is the ability to envision long-term issues but at the same time, have the desire for closure with the current situation in order to move on.

4.2.3. **Analytic *and* Synthetic:** A big-picture perspective may be associated with the ability to be synthetic, and to be able to bring together and integrate different pieces of a puzzle. However, a systems engineer also needs to be analytic and to be able to break down the big picture into smaller pieces on which others can focus and work. To do this effectively, a systems engineer needs to be able to operate at multiple levels (e.g., component, sub-system, system, system-of-systems) and multiple dimensions (e.g., various technical disciplines and stakeholder perspectives).

4.3. **Adaptability**: *The overall ability to deal with ambiguity and uncertainty, this involves the abilities to be open-minded, understand multiple disciplines, deal with challenges, and the ability to take rational risks*. By definition, experts possess proficiency in a specific area, which is their 'comfort zone'; and they typically do not prefer going outside that circle or comfort zone. Such experts provide value to the organization by contributing their expertise in those focused areas. However, systems engineers tend to show an ability to broaden their comfort zones, and go beyond their current boundaries and they are also comfortable doing this.

Note: Previously this category was titled, "Flexible Comfort Zone". The Helix team received feedback that "Flexible Comfort Zone" sounded much more like a personal characteristic than a skillset. In looking at the data, individuals who talked about "flexibility " or a "flexible comfort zone", what they were really talking about was the ability for an individual to adjust successfully to change. The team reviewed many definitions of flexibility, which tended to focus on the ability to be modified, whereas adaptability more accurately reflects the quality of being able to adjust to new conditions. The Helix team believes that with this change, confusion on overlap between this proficiency category and personal characteristics will be alleviated and that, in addition, the new title more accurately reflects the skills to which interviewees were referring.

4.4. **Abstraction**: *The ability to filter out and understand the critical bits of information at the right level and to make relevant inferences*. And even with that filtered information, systems engineers need to know when to use or not use pieces of information. Such abstraction also enables systems engineers to connect and extract meaning from different streams of information; for example, to tie together information that subject matter experts of two different disciplines are providing.

4.5. **Foresight and Vision**: *The ability to foresee the remaining lifecycle of the system, the impact of current decisions, and to mentally simulate possible scenarios*. Every decision or change is likely to have an impact beyond the current confines of time or space. Particularly in early stages of a system lifecycle, and in the development of a new or unfamiliar system, foresight is a key value that systems engineers provide.

5.1.5 AREA 5: INTERPERSONAL SKILLS

The fifth proficiency area is *Interpersonal Skills*. Almost by definition, systems engineers do not just work by themselves at their desks all day – they interact with others. Irrespective of any formal leadership roles they may or may not play, a systems engineer is expected to be proficient in a number of interpersonal skills. While specialty engineers may be responsible for developing specific aspects of the system, systems engineers are responsible for coordinating across all of these engineers. Hence, interpersonal skills are more critical to systems engineers than they are to specialty engineers. The specific categories contained within this proficiency area are listed below:

5.1. Communication: Communication is critical for systems engineers since they interact with a variety of people, and is a broad category covering a wide variety of related skills and abilities. Often they are an important link between individuals and groups, both internal and external to the organization – most importantly, the customers and end-users of the system being developed. Systems engineers need the ability to clearly express their thoughts and perspectives to establish a shared common understanding.

5.1.1. Audience: Systems engineers need to communicate with a variety of direct and indirect audiences: customers; subject matter experts; program managers; vice presidents; directors; specialty engineers; problem owners; technical teams; contractors; decision makers; system testers; and others working on or with the project.

5.1.2. Content: The variety of content that systems engineers need to communicate can be broadly divided into three types, based on the audience they are communicating with:

1. **Technical**: Communications with disciplinary and specialty engineers and subject matter experts involve high technical content. But communications of technical issues to managers, end-users, and others who may not be interested in or who may be confused by all the technical detail, involves adequate abstraction of the technical content.

2. **Managerial**: Systems engineers often provide project status to managers and supervisors and cost-schedule constraints and expectations to technical personnel.

3. **Social**: Systems engineers need to maintain an amicable environment within a team and to interact with others in a courteous manner. Such interactions involve communications that are neither technical nor managerial in nature.

5.1.3. Mode: Communicating the intended content to the target audience is done through a number of different modes:

1. **Oral:** This takes various forms, depending on the audience and context. It could be one-on-one, or as part of a team, in person, or remotely.

2. **Presentation**: A special form of communications is the ability to stand in front of an audience and to deliver a presentation using appropriate aids. Further, during presentations, systems engineers tend to represent others who may not be in the room: they present customer needs and requirements to others in the absence of customers, and they present design decisions and system related issues to customers in the absence of designers.

3. **Writing and Documentation:** Written communication skills are equally critical for

systems engineers; the scale, audience, and objective of the written artifact also matter. It could range from a short email to communicate status, to a detailed test plan, to internal documentation supporting a project decision, to design documents being submitted for review.

5.2. **Listening and Comprehension:** The ability to listen to others' points of views and perspectives, and to comprehend and internalize the message accurately. For systems engineers, listening begins with the customer to understand their real needs and ensure that these needs get translated into requirements. In a team environment, systems engineers need to listen to the views and perspectives being offered: from designers, subject matter experts, and others.

5.3. **Working in a Team**: Systems engineers tend to be part of many teams during the lifecycle of the system; further, systems engineering by itself is typically not performed by an individual, but rather by a team. Hence, team dynamics and synergy are key to the functioning of a systems engineer.

5.4. **Influence, Persuasion, and Negotiation**: It is critical for every systems engineer, not just those in formal leadership positions, to have the skills needed to make a point and to successfully obtain buy-in. In many situations, systems engineers contribute a perspective that is different from that of others: a focus on the overall system and on customer's needs. In such situations, it requires influence, persuasion, and negotiating skills for systems engineers to enable others to see the bigger picture on which they need to focus.

5.5. **Building a Social Network:** A systems engineer needs to be a 'people person', and build a social network of professional acquaintances. Such a network becomes a valuable resource for systems engineers to tap into, because they are not expected to know answers to all problems, but rather be able to find someone who has the expertise and ability to solve the problem.

5.1.6 Area 6: Technical Leadership

The sixth and final *Atlas* proficiency area is *Technical Leadership*. It is common and natural for systems engineers to play leadership roles at many levels within an organization. The specific categories contained within *Technical Leadership* are listed below:

6.1. **Building and Orchestrating a Diverse Team**: *The ability to identify, build, and effectively guide or coach a team comprising individuals with diverse expertise, perspectives, and personalities*. While organizational titles may vary, it is most often a systems engineer who is the leader of the team that is charged with delivering the system. The systems engineer needs to fully know each of the team members: their strengths, weaknesses, capacities, capabilities, limitations, personalities, expertise, and working styles. The systems engineer plays the roles of coach, guide, and teacher to develop the team's capabilities and to orchestrate it to perform the required tasks. Individual leadership styles could vary, but the overall objective of is to empower the team, to instill confidence, and to help them to deliver the solution and to be successful. Another key aspect of handling a team is the ability to delegate – the leader needs to build enough trust in the team to be able to delegate with confidence.

6.2. **Balanced Decision Making and Rational Risk Taking:** Solving a problem requires a systems engineer to take a number of balanced decisions considering a variety of factors, constraints, perspectives, and objectives; as well as the implications of these decisions and their scope of impact. An additional challenge is that most often, all the required information may not be

readily available. The ability to make such decisions also requires the systems engineer to be comfortable in dealing with ambiguity and uncertainty and to be able to take rational, calculated risks.

6.3. **Guiding Diverse Stakeholders:** This includes the ability to manage all the internal and external stakeholders, and to keep the team focused on their needs, especially those of the end user or customer. The systems engineer is uniquely positioned to interact with many stakeholders of the system – both external and internal to the organization. Being this "touch point" person, the systems engineer needs to deal with multiple personalities, behaviors, organizations, and cultures.

> *Note: In Atlas 1.0, this was "Guiding Stakeholders with Diverse/Conflicting Needs". The Helix team received feedback that the previous title made overlaps with other elements of Atlas unclear. For example, as titled, people reported being confused about the distinction between this proficiency and the value of guiding diverse teams. Proficiencies are specific skills sets while Values are the end-stage value that systems engineers provide using a number of proficiencies. While there is a clear relationship between these, the team believes the updated language will help users avoid any confusion. Likewise, the previous title seemed to overlap with the next category, "Conflict Resolution and Barrier Breaking." Again, the change in title is intended to improve this.*

6.4. **Conflict Resolution and Barrier Breaking**: Conflicts are bound to rise in a variety of scenarios – within the team; within the organization – between the technical side and business side of the organization; as well as with outside the organization. As a leader, the systems engineer must resolve these conflicts while keeping the system goals in mind. In some cases, conflicts arise due to the existence of barriers, which may be related to the organizational culture, processes, team personalities, or other situations that could prevent an individual or team from getting their work done. The systems engineer needs the ability to break these barriers.

6.5. **Business and Project Management**: Depending on the way roles and titles are defined within an organization, a systems engineer's responsibilities may overlap with what may be seen as 'project management' responsibilities. Even if there is no overlap, a systems engineer is expected to handle a variety of business and project management activities including accounting, budget, cost estimation, schedule, work breakdown, and profit. The systems engineer must also be cognizant of the business impact of technical decisions that are taken.

6.6. **Establishing Technical Strategies:** Systems engineers must fearlessly and creatively guide the establishment of new capabilities and transformations (e.g., to migrate to Cloud Infrastructure, or to establish a new information service architecture, or to enable transition to a DEVOPS model). Senior systems engineers need to be able to support the organization in the development of overarching technical directions and support the development of technical roadmaps that establish a vision to support the strategy.

6.7. **Enabling Broad Portfolio-Level Outcomes:** Along with the development of strategies to guide strategic technical investments, systems engineers should provide the broad perspective necessary to enable technical success not only on individual projects but across projects and programs to enable advancement across the technical portfolio.

5.2 Tailoring the Proficiency Framework

As demonstrated in Table 5, above there is a clear expectation that some tailoring will occur for proficiency assessment to maximize its utility. This is true for both individuals and organizations. Individuals may tailor the model specifically to what they have done – but should be mindful that all of the areas they have not touched are possible areas for future exploration. Organizations, likewise, could tailor the model before distributing it to the workforce, so that only areas that are deemed critical to the organization are captured. For example, some of the natural science foundations may not be common in a given domain and some disciplines or technologies will be considered more relevant than others. It is important to remember that tailoring may not be specific to just an organization, but also to specific programs or systems. For example, an organization that engineers financial IT systems as well as critical infrastructure systems may have different expectations and needs for those different domains.

Table 6 provides two examples of how the proficiency model could be tailored for an organization, based on the primary systems domain for each organization. Note that where <no tailoring> is listed, this indicates that the Helix team expects that either an organization will be able to use the proficiency model exactly as defined, with no tailoring required, or that for purposes of this example, no specific tailoring has been identified.

Table 6. Tailoring the Atlas Proficiency Framework

Area	Category	Company 1: Defense Aerospace	Company 2: Medical Devices
1. Math / Science / General Engineering	**1.1.** Natural Science Foundations	Physics considered most critical	Chemistry and Biology considered most critical Physiology added as a Foundation
	1.2. Engineering Fundamentals	<no tailoring>	<no tailoring>
	1.3. Probability and Statistics	<no tailoring>	<no tailoring>
	1.4. Calculus and Analytical Geometry	Both are considered critical	Considered less critical than Probability & Statistics
	1.5. Computing Fundamentals	Considered less critical than the other categories	Considered critical for integration with Electronic Health Records (EHRs)
	1.6. Social Sciences		Psychology
2. Systems' Domain & Operational Context	**2.1.** Principal and Relevant Systems	Air-breathing jet engines Military aircraft	Magnetic Resonance Imaging (MRI) X-Ray Computerized Tomography (CT)
	2.2. Familiarity with Principal System's Concept of Operations (ConOps)	Expectations about the level of familiarity may differ (e.g. understanding basic in-flight operations)	Expectations about the level of familiarity may differ (e.g. actual experience in a clinical setting to understand use cases, how system fits within the healthcare environment, where its use may fit in an overall process,

Area	Category	Company 1: Defense Aerospace	Company 2: Medical Devices
			etc.)
	2.3. Relevant Domains	Aerospace	Healthcare
	2.4. Relevant Technologies	Radar Sonar Navigation Systems	MRI X-Ray CT
	2.5. Relevant Disciplines and Specialties	Mechanical Engineering Electrical Engineering Aerospace Engineering Software Engineering Thermodynamics Aerodynamics Ergonomics	Electrical Engineering Mechanical Engineering Biomedical Engineering Software Engineering Ergonomics Radiation Safety
	2.6. System Characteristics	System level design with understanding of the system of systems in the operational environment	Systems of systems level design enabling integration with other medical devices and healthcare IT systems
3. Systems Engineering Discipline	**3.1.** Lifecycle	• V-lifecycle approach emphasized • Organization not involved in in-service operation and maintenance (full handoff after delivery)	• Spiral/Incremental Development lifecycle model emphasized • Organization heavily involved in in-service operation and maintenance
	3.2. Systems Engineering Management	<no tailoring>	<no tailoring>
	3.3. SE Methods, Processes, and Tools	• Heavy emphasis on modeling and simulation • Emphasis on operational safety	• Heavy emphasis in optimization for patient safety
	3.4. Systems Engineering Trends	• Model Oriented Systems Engineering	<no tailoring>
4. Systems Mindset	**4.1.** Big-Picture Thinking	<no tailoring>	<no tailoring>
	4.2. Paradoxical Mindset	• Balance of Methodical and Creative heavily weighted	• Paradoxical mindset heavily weighted
	4.3. Adaptability	<no tailoring>	<no tailoring>
	4.4. Abstraction	<no tailoring>	<no tailoring>
	4.5. Foresight and Vision	<no tailoring>	<no tailoring>

Area	Category	Company 1: Defense Aerospace	Company 2: Medical Devices
5. Interpersonal Skills	**5.1.** Communication	<no tailoring>	<no tailoring>
	5.2. Listening and Comprehension	<no tailoring>	<no tailoring>
	5.3. Working in a Team	<no tailoring>	<no tailoring>
	5.4. Influence, Persuasion, and Negotiation	<no tailoring>	<no tailoring>
	5.5. Building a Social Network	<no tailoring>	<no tailoring>
6. Technical Leadership	**6.1.** Building and Orchestrating a Diverse Team	<no tailoring>	<no tailoring>
	6.2. Balanced Decision Making & Rational Risk Taking	<no tailoring>	Risk is viewed negatively by this highly safety-conscious organization; this becomes focused on decision making.
	6.3. Guiding Diverse Stakeholders	<no tailoring>	<no tailoring>
	6.4. Conflict Resolution & Barrier Breaking	<no tailoring>	<no tailoring>
	6.5. Business and Project Management Skills	• Project management is treated as a distinctly separate discipline from systems engineering in this organization. There is cultural pressure not to include this as a "systems engineering" proficiency.	<no tailoring>
	6.6. Establishing Technical Strategies	• N/A (Systems engineers do not set the technical strategy for the organization)	• Only expected for senior systems engineers
	6.7. Enabling Broad Portfolio-Level Outcomes	• N/A (Systems engineers do not set the technical strategy for the organization)	• Only expected for senior systems engineers

Table 6 is only a basic example, but demonstrates that tailoring can include the identification of specific proficiencies that are of critical interest to the organization – particularly in Proficiency Areas 1 and 2, which are expected to be heavily tailored – and the emphasis or de-emphasis of categories based on the organizational context. The examples for categories 6.6 and 6.7 also demonstrate that the organization can help to set expectations about categories that are critical only at certain seniority levels.

5.3 Proficiency Assessments

One of the areas that has proven more difficult than expected for the Helix team is the development of a rubric to guide assessment of proficiencies. The team has helped over 100 individuals conduct self-assessments and had exploratory conversation around these assessments, but the primary roadblock to this has been that individuals struggle to explain skills versus how they attained them. For example, if an individual said that they were a 6 out of 10 for "Systems Engineering Discipline", the team would ask what that "6" really meant. The answers would often be something like this: Well, I've been doing systems engineering for 5 years and I've seen most of the lifecycle and I am good with the tools we utilize here. Note that "I've seen most of the lifecycle" – an aspect of their career path – is different from "I am able to provide clear value and leadership at any stage of the lifecycle." When the team probed further, individuals simply did not have the vocabulary to describe precisely the differences between a "5 out of 10" and a "7 out of 10".

In their work to be published in 2018, Pyster, Hutchison, and Henry tackled this in a different way. They identified a comparable proficiency scale which is somewhat generic – utilizing broad descriptions for a level of proficiency – rather than trying to tailor a specific definition for every single topic. This is adapted from a rubric developed by the National Institutes of Health (NIH), the "NIH Proficiency Scale is an instrument used to measure one's ability to demonstrate a competency on the job. The scale captures a wide range of ability levels and organizes them into five steps; from 'Fundamental Awareness' to "Expert'." Pyster et al. have adapted this to apply to the *Atlas* framework, translating the levels into a 5-point scale for each of use. (2018, in print) This is illustrated in Table 7.

Table 7. Proficiency Levels (adapted from Pyster et al. 2018, in print, used with permission)

#	Level	Level Description
1	**Fundamental Awareness**	Individual has common knowledge or an understanding of basic techniques and concepts. Focus is on learning rather than doing.
2	**Novice**	Individual has the level of experience gained in a classroom or as a trainee on-the-job. Individual can discuss terminology, concepts, principles, and issues related to this proficiency, and use the full range of reference and resource materials in this proficiency. Individual routinely need help performing tasks that rely on this proficiency.
3	**Intermediate**	Individual can successfully complete tasks relying on this proficiency. Help from an expert may be required from time to time, but the task is usually performed independently. The individual has applied this proficiency to situations occasionally while needing minimal guidance to perform it successfully. Individual understands and can discuss the application and implications of changes in tasks relying on the proficiency.
4	**Advanced**	Individual can perform the actions associated with this proficiency without assistance. The individual has consistently provided practical and relevant ideas and perspectives on ways to improve the proficiency and its application and can coach others on this proficiency by translating complex nuances related to it into easy to understand terms. Individual participates in senior level discussions regarding this proficiency and assists in the development of reference and resource materials in this proficiency.

#	Level	Level Description
5	**Expert**	Individual is known as an expert in this proficiency and provides guidance and troubleshooting and answers questions related to this proficiency and the roles where the proficiency is used. Focus is strategic. Individual have demonstrated consistent excellence in applying this proficiency across multiple projects and/or organizations. Individual can explain this proficiency to others in a commanding fashion, both inside and outside their organization.

During some of the Helix interviews in 2015-2017, interviewees were asked to self-evaluate their level of proficiency based on the *Atlas* proficiency model, at the Area level. Generally, interviewees evaluated themselves on a level of 1 to 10, where 1 was 'least proficient' and 10 was 'most proficient'. This was a subjective scale and hence when someone placed themselves at an 8 for a proficiency area, for example, it was based on their personal interpretation on what it meant.

Interviewees were asked to evaluate their proficiencies at two points in time: (1) at the time of the interview, and (2) at the start of their career. This enables a proficiency profile to be plotted, as illustrated in Figure 8.

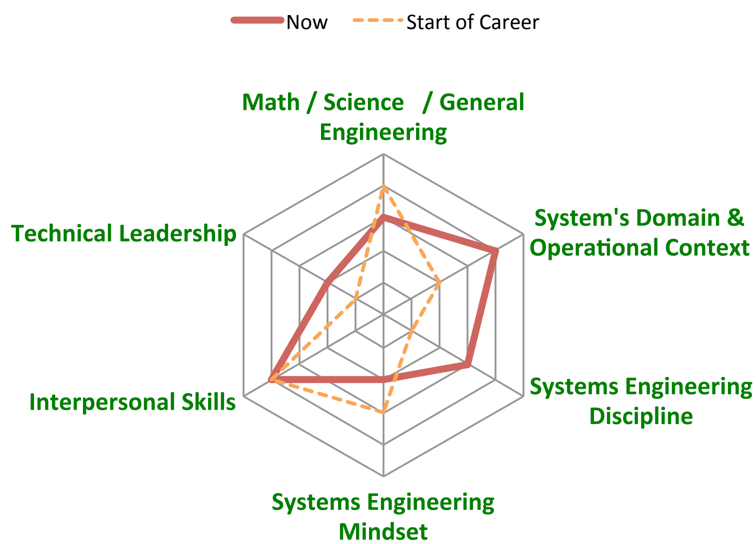

Figure 8. Proficiency Profile of an Individual

The proficiency profile is not meant to be exact since the self-evaluations are subjective, and individuals may have over-evaluated or under-evaluated themselves. Also, 'Start of Career' could be as recent as five years ago for one individual or twenty-five years ago for another. However, this exercise enables a discussion around the relative strengths in specific proficiencies; how proficiency levels changed over time; and what factors or forces caused or enabled those changes.

The primary intent of *Atlas* is not to just understand the current state of effective systems engineers, but to support the development of future systems engineers who will be effective. From a proficiency perspective, it would mean setting target levels for proficiency areas, as illustrated in Figure 9.

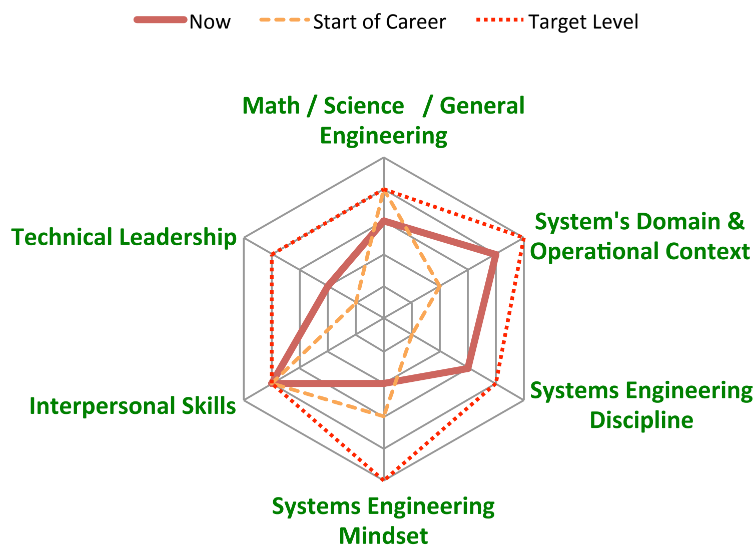

Figure 9. Proficiency Profile with Target Levels

5.3 Comparing the *Atlas* Proficiency Model to the Draft INCOSE Competency Model

In 2015, the UK Chapter of the International Council on Systems Engineering (INCOSE) produced the "Systems Engineering Competency Framework." This framework has been heavily referenced and led to an international INCOSE initiative to develop a core competency model. The latest version of the competency model, v. 0.75, was released in 2017 and the final version, 1.0, is anticipated for release in 2018. The competency model for INCOSE is being generated by the Competency Working Group, a team of systems engineering practitioners from around the world. This is a different approach than the *Atlas* model, which was based on grounded theory data collection. Nonetheless, there are areas of overlap between the models. To help inform community discussion, Table 8 provides an overview of the overlap between the INCOSE v. 0.75 competency model and the *Atlas* 1.1 proficiency model.

Table 8. Comparison of *Atlas* Proficiency model and INCOSE (draft) 0.75 Competency Model

INCOSE Model		Atlas 1.1 Model
Competence Group	Competence Area	
Core SE Principles	**Systems Thinking**	Aligns with Big Picture Thinking Category, Systems Mindset Area
	Lifecycles	Aligns with Lifecycles Category, Systems Engineering Discipline Area
	Capability Engineering	Defined in the 0.75 Competency Model as an appreciation for the super system of which the system is a part, this aligns with the Big Picture Thinking Category, Systems Mindset Area.

INCOSE Model		Atlas 1.1 Model
Competence Group	Competence Area	
	General Engineering	Defined in the 0.75 Competency Model as basic scientific and engineering knowledge and its application, this aligns with the Natural Sciences and Engineering Fundamentals Categories in the Math/Science/General Engineering Area.
	Critical Thinking	This aligns with the Balanced Decision Making and Rational Risk Taking Category in the Technical Leadership area, though the Atlas model does not state critical thinking explicitly.
	Systems Modeling and Analysis	This aligns with the Systems Engineering Analytics and Model Oriented Systems Engineering Topics in the Systems Engineering Trends Category as well as the Methods, Processes, and Tools Category in Systems Engineering Discipline.
Professional Competencies	**Communications**	This aligns with the Communications Category in Interpersonal Skills area.
	Ethics and Professionalism	This is not incorporated into the Proficiency model of Atlas but is rather separated as the Personal Enabling Characteristic of Professionalism and Respect.
	Technical Leadership	Though the title in the INCOSE model is Technical Leadership, this does not align with the Technical Leadership area of Helix. Defined as broad technical domain knowledge, engineering instinct, problem solving, creativity, and the leadership and communication skills needed to develop new missions and systems, these skills are distributed among a variety of Atlas areas, including System's Domain and Operational Context, Systems Engineering Discipline, Math/Science/General Engineering, and Technical Leadership.
	Negotiation	Aligns with the Influence, Persuasion, and Negotiation category of the Interpersonal Skills area.
	Team Dynamics	Aligns with the Building and Orchestrating a Diverse Team category of the Technical Leadership area.
	Facilitation	Aligns with Influence, Persuasion, and Negotiation category of Interpersonal Skills area.
	Emotional Intelligence	Emotional Intelligence is not covered in the *Atlas* proficiency model, though it would support a number of the topics and skills. The closest item that aligns in *Atlas* is the Self-Awareness personal enabling characteristic.

INCOSE Model		Atlas 1.1 Model
Competence Group	Competence Area	
	Coaching and Mentoring	In *Atlas*, coaching and mentoring are classified as a *Force* for improving systems engineering capabilities. The ability for senior systems engineers, especially, to coach and mentor other systems engineers is highlighted, but not incorporated into the Proficiency model.
Technical Competencies	**Requirements Definition**	In Atlas, these competencies are covered in a few areas. In the Proficiency model, they are included in the Lifecycle category - which is not only awareness and understanding of the lifecycle, but the ability to do the work required in each phase of the lifecycle. In addition, the Roles defined in Atlas are specifically around these types of activities. The Design For . . . area of the INCOSE competency model is a bit different and is not included in the Atlas model. In the Helix data, though systems engineers dealt with constraints such as reliability, security, safety, etc., they often reported that they worked with subject matter experts who provided the key insights. It is for this reason that it is not reflected in Atlas as a systems engineering capability.
	System Architecting	
	Design for…	
	Integration	
	Interfaces	
	Verification	
	Validation	
	Transition	
	Operation and Support	
SE Management Competencies	**Planning**	Systems Engineering Management is a category within the Systems Engineering Discipline area of Atlas. The topics included in this category align well with the areas listed in the INCOSE model. In addition, several of these are also called out specifically as systems engineering Roles in Atlas, such as Configuration Manger and Information Manger.
	Monitoring and Control	
	Decision Management	
	Concurrent Engineering	
	Business & Enterprise Integration	
	Acquisition and Supply	
	Information Management	
	Configuration Management	
	Risk and Opportunity Management	
Integrating Competencies	**Project Management**	Aligns with the Business and Project Management category of the Technical Leadership area in Atlas.
	Finance	These are not called out specifically in *Atlas,* but instead were viewed in the dataset as specific design considerations that systems engineers might deal with - the same as systems
	Logistics	

INCOSE Model		Atlas 1.1 Model
Competence Group	Competence Area	
	Quality	engineers would have to consider the domain, the operational context, and the stakeholders.

Overall, the Helix team found that though the different approaches taken led to different grouping of knowledge, skills, and abilities, the INCOSE 0.75 Competency Model and the *Atlas* proficiency model aligned well. There are different areas of emphasis, with skills sometimes being more distributed in one model than other, but overall there is good alignment.

The following are *Atlas* proficiencies for which the team could find no clear analog in the INCOSE 0.75 competency model – though again, these may be implied in or distributed across the INCOSE model, just as some INCOSE competencies are distributed across several proficiencies in *Atlas*.

- Probability and Statistics
- Calculus and Analytical Geometry
- Computing Fundamentals
- Adaptability
- Abstraction
- Foresight and Vision
- Paradoxical Mindset
- Building a Social Network
- Establishing Technical Strategies
- Enabling Broad Portfolio-Level Outcomes

6 Forces That Enable Systems Engineers to Grow

This section is has been updated from Atlas *1.0 with additional details that were previously contained in the associated technical report. The details have not changed since* Atlas *1.0.*

The three most important forces that significantly impact the proficiency of systems engineers are *Experiences*, *Mentoring*, and *Education & Training*, in that order. These forces are generated by a combination of personal and organizational initiatives. The application of these forces is the primary way by which proficiencies of an individual are developed, as illustrated in Figure 10 below.

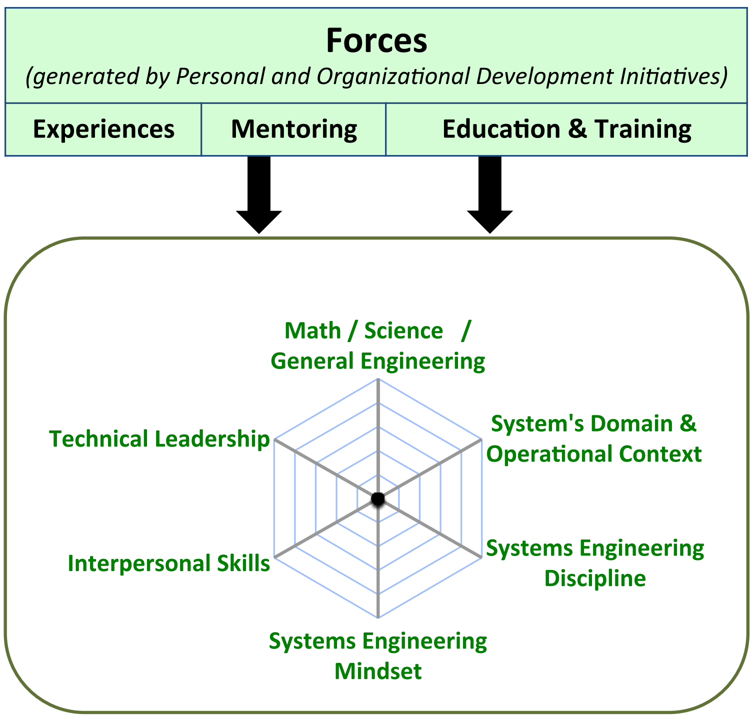

Figure 10. Forces and Proficiency

6.1 Force 1: Experiences

Experiences are considered the most critical factor contributing to the development of proficiencies and to the overall growth of systems engineers. However, it is the characterization of these experiences that provides insight into how they impact proficiencies over time. Considering experiences as a force, each of these dimensions contributes to increasing one or more areas of proficiency. Experiences can also impact the personal characteristics of an individual. *Experiences*, as considered in *Atlas*, includes experiences along the following characteristics:

- **Relevance:** Every experience cannot be considered to be relevant to the development of systems engineers. A 'relevant' position is one that enables a systems engineer to develop the

proficiencies critical to systems engineering. A 'systems engineering' position is one where the individual's primary focus was on SE activities.

- **Position:** Every systems engineer who is employed at an organization fills a position that is established by the organization; that organization also defines the roles and responsibilities to be performed. Helix considers position as a 'unit of measure' for experience, since most of the characteristics of experience is in the context of the position that is being held.
- **Chronological Time:** The amount of time spent in any particular position or in performing a role.
- **Number of Organizations:** The number of different organizations that an individual has worked at, not counting internal movement within an organization across departments or divisions, reflects the variety of experiences that one may possess. In large corporations that have multiple business units, or in situations where there are mergers and acquisitions, this number may not be a good indicator of the variety of experiences.
- **Organizational Type:** There are many differences in the general characteristics of an organization based on its sector. In *Atlas*, three organizational sectors are identified: government, industry, and FFRDC. Academic organizations could also be included, those these were not the focus of the Helix work.
- **Organization Domain:** Some organizations focus primarily on one domain, while others work within a variety of domains. The primary domain can have important impacts on the organizations culture (see Section 7).
- **Roles:** The 15 roles identified in *Atlas* are described in Section 4.
- **Lifecycle Phases:** The lifecycle phases used in *Atlas* are reflected in Table 9. The titles and descriptions of lifecycle phases or stages may vary across different systems engineering processes and frameworks available in literature or in use at an organization.

Table 9. Definition on lifecycle phases according to SEBoK (BKCASE Authors, 2015)

Lifecycle Phase	Definition
Concept Definition	A set of core technical activities of SE in which the problem space and the needs of the stakeholders are closely examined. This consists of analysis of the problem space, business or mission analysis, and the definition of stakeholder needs for required services within it.
System Definition	A set of core technical activities of SE, including the activities that are completed primarily in the front-end portion of the system design. This consists of the definition of system requirements, the design of one or more logical and physical architectures, and analysis and selection between possible solution options.
System Realization	The activities required to build a system, integrate disparate system elements, and ensure that a system both meets the needs of stakeholders and aligns with the requirements identified in the system definition stage. This includes integration, verification, and validation (IV&V)
System Deployment and Use	A set of core technical activities of SE to ensure that the developed system is operationally acceptable and that the responsibility for the effective, efficient, and safe operations of the system is transferred to the owner. Considerations for deployment and use must be included throughout the system life cycle. Activities within this stage include deployment,

Lifecycle Phase	Definition
	operation, maintenance, and logistics
Product and Service Life Management	Deals with the overall life cycle planning and support of a system. The life of a product or service spans a considerably longer period of time than the time required to design and develop the system. This stage includes service life extension, updates, upgrades, and modernization, and disposal and retirement. The organizations in the current sample are primarily concentrated on new development, so this is a very under-represented aspect of the life cycle.
Systems Engineering Management	Managing the resources and assets allocated to perform SE activities. Activities include planning, assessment and control, risk management, measurement, decision management, configuration management, information management, and quality management. These activities can occur at any point in the systems engineering lifecycle.

- **Systems:** There are many aspects to the types of systems on which a systems engineer could work. Working across these different categories provides valuable experience to an individual systems engineer.
    - **Domain:** This is the primary area of application for the systems being worked on. However, there are many domain categorizations; some domains also relate to industry sectors.
    - **Type:** Product systems, service systems, and enterprise systems are three major types of systems, depending on the nature and composition of the system of interest. System of systems is another paradigm in systems engineering, and could be a combination of one or more types of systems.
    - **Level:** A systems engineer could work on various levels of a system: component/element, subsystem, system, and platform or system of systems.

For additional details on how the Force of experiences impact systems engineers' proficiencies, see the *Atlas Career Path Guidebook*. (SERC-2018-TR-101-C)

6.2 Force 2: Mentoring

Mentoring (or mentorship) is a relationship between two individuals: a mentor possesses more experience and knowledge and shares these with a mentee for the mentee's personal development. The effectiveness and derived value of the mentoring relationship is dependent on the individuals involved, but is also influenced by the organization which derives value out of a mentoring relationship as well.

6.2.1 What is Mentoring?

Mentoring means different things to different individuals and in different organizations. Common characteristics of mentoring are discussed below.

- Two individuals are involved in a mentoring arrangement: a mentor and a mentee (also referred to as a protégé).

- The mentor is usually *senior* when compared to the mentee in age, experience, and/or expertise.
- Primarily, the mentor *gives* and the mentee *receives*.
- The mentor-mentee relationship is a many-many relationship: a single mentor can have multiple mentees, and a single mentee can have multiple mentors – concurrently or spread over time.
- Mentor-mentee interactions typically happen over an extended period of time at varying frequencies.

There are also some differences and contradictions in the understanding of mentoring.

- Some use the term mentoring to describe any interaction with any co-worker in the organization that would provide any advice or guidance to handle the problem at hand.
- Some consider mentors to be synonymous with subject matter experts (SMEs) who are consulted for their expertise on an as-needed basis only. In contrast, some consider it mentoring only if the mentor is a senior person, and only if there are regular interactions between the mentor and mentee over an extended period of time.
- When the mentor and the mentee are of the same seniority in terms of age, years of experience, or level of expertise, some still consider it to be a mentoring relationship, while some others consider it to be a peer-peer relationship and not a mentoring relationship.
- Some distinguish between the concepts of coaching and mentoring: coaching is related to providing advice and guidance on solving a specific technical problem, while mentoring on the other hand, has neither a set beginning or end to the relationship, nor is related to a specific event.

6.2.2 Mentoring Arrangements

Mentoring arrangements can either be formal or informal, depending on the level of engagement of the organization in establishing and sustaining the mentoring relationship. The two types of mentoring arrangements may be summarized as below:

- **Formal**: The organization plays an active role in establishing the mentor-mentee relationship, and also lays down guidelines for maintaining that relationship. Usually, organizations require that objectives and expectations for the mentor and the mentee be stated explicitly. The relationship and its progress tend to be monitored by the organization.
- **Informal**: The participating individuals establish the mentor–mentee relationship by themselves: either a mentor adopts a mentee or a mentee seeks out a mentor, and the relationship is established. Formal objectives or expectations are usually not stated explicitly, but it is considered good practice to establish these in some form at the start of the relationship. The organization plays a less active role in informal mentoring. It is upon the mentor and the mentee to establish and drive the relationship.

6.2.3 Mentoring Focus

Depending on what the mentoring is about, interviewees mentioned three types of mentoring:

- **Career Mentoring:** The mentor provides advice on career-related issues: helps identify career goals and the paths leading to that goal. The mentor could be from another group or division in the organization. Mentees are also groomed on management and leadership related topics.
- **Technical Mentoring:** The mentor typically provides advice on the technical details of the system being engineered. The mentor teaches lessons that are typically not found in textbooks and provides crucial insights on technical tools and processes. The mentor also acts as a subject matter expert, answering questions mentees might have on the subject, the system, or the program.
- **Organizational Mentoring:** While closely related to career mentoring, in organizational mentoring the mentor provides information about the organization: its culture, its procedures, and its policies. This is especially critical to a new employee.

6.2.4 Benefits of Mentoring

In any typical mentoring arrangement, the mentor 'gives' and the mentee 'receives'. Therefore, such an arrangement is expected to be most beneficial to the mentee. However, there are benefits to the mentors as well. In addition, the organization also stands to benefit. Whenever an organization establishes a formal mentoring initiative, it usually expects to derive some benefit out the mentoring arrangements. However, the benefits to the mentee, to the mentor, or to the organization are conditional, and should not be taken for granted.

- **Benefits to Mentees**: The mentee gains significantly through mentoring. Most interviewees identified mentoring as a critical factor that increases the effectiveness of systems engineers. The biggest benefit to mentees of mentoring is the relationship they establish with their mentors over the span of their careers; most other benefits of mentoring are enabled through the mentor. Through their mentors, employees often get exposed to opportunities within the organization that may not be visible otherwise. During mentoring, mentees often receive important lessons from their mentors, which have made a significant impact in their careers. Finally, mentoring enables a mentee to build a strong professional network.
- **Benefits to Mentors:** Though the mentee stands to benefit the most, the mentor also benefits by mentoring, which tends to motivate the mentor to engage in a mentoring a relationship. Many considered mentoring to be an important part of their jobs; helping rising stars and teaching younger engineers what to do was motivation enough for most mentors. In organizations where mentoring is acknowledged, mentors get recognized for their efforts, for example in annual performance evaluations. Some mentors considered mentoring to be a means of reducing their workload when a mentee is able to take responsibility for a portion of the work. Finally, mentoring can be a critical way to groom a successor. This was particularly heard from senior systems engineers, but could be relevant at any stage in the career.
- **Benefits to Organization**: Effective mentoring not only benefits the mentees and mentors involved in the relationship, but also the workforce as a whole. When this happens, the organization at large benefits as well. Good mentoring was seen as one of the most efficient

ways enable effective knowledge transfer from the senior members of the workforce to more junior members. Through the feedback from mentors, organizations can also identify high-potential engineers who are being mentored. Effective mentoring can significantly reduce the time taken for new employees to get oriented to their jobs, making them effective more quickly. Effective mentoring was also seen as a mechanism for improving employee retention; when an individuals feel they have someone "in their corner" who is helping them on the job and shepherding their careers, they are more likely to feel valued and less likely to look for opportunities outside the organization.

6.3 Force 3: Education & Training

Education plays two key roles in the development of systems engineers:

1. It provides the foundation knowledge to support engineering-related work. Typically, this takes the form of undergraduate education in an engineering discipline, technical field, or physical science.
2. Graduate level education is an avenue to develop more advanced skills, explore more in-depth knowledge, and help systems engineers grow as they move through their careers.

In addition to formal academic programs leading to undergraduate and graduate degrees, there are graduate certificates that individuals obtain, in an area that is closely related to their work. Some systems engineers go on to obtain doctoral degrees as well.

Systems engineers typically start their careers after obtaining an undergraduate degree, while graduate degrees may be obtained immediately after an undergraduate program or after a few years of professional work. Any formal degree directly improves proficiency in the relevant areas and categories. Any undergraduate degree in engineering typically provides much of the *Math/Science/General Engineering* proficiency in addition to the relevant categories under the *Systems' Domain & Operational Context* proficiency area. Graduate degrees add to relevant proficiencies; much of the formal systems engineering education happens at the graduate level.

While academic programs are typically offered by a university, there are a number of tailored training programs that organizations offer their employees. These trainings are more focused on building specific skills that are required for them to perform their work and are typically offered short-term. The topics vary widely across organizations, with some training focused on the technical aspects of systems development, other training focused on organization-specific approaches and processes, and still other training focused on leadership or interpersonal skills. Each type of training has a role in the development of proficiency.

Among the six proficiency areas in *Atlas*, *Math/Science/General Engineering*, *System's Domain & Operational Context*, and *Systems Engineering Discipline* may be considered to be 'hard' proficiencies at large, while *Systems Engineering Mindset*, *Interpersonal Skills*, and *Technical Leadership* may be considered to be 'soft' proficiencies at large. Formal education typically improves the hard proficiencies, but training could improve both hard and soft proficiencies.

In general, education or training results in an initial, single increase in proficiency. Additional changes over time are then the result of applying the knowledge or skills gained through this force in a real-world setting; i.e., through experiences utilizing the outputs of the education or training.

Characteristics that would be identified for relevant Education and Training would include:

- Type (education or training)
- Duration
- Date/Type of Completion (graduation date for an academic degree, course completion date for a single educational or training course)
- Subject matter covered
- Expected and/or Actual Outcomes, particularly in the context of expected changes to a systems engineer's proficiency after completion.

For additional details on how the Force of education and training impact systems engineers' proficiencies, see the *Atlas Career Path Guidebook*. (SERC-2018-TR-101-C)

7 Personal and Organizational Characteristics That Impact Systems Engineers' Effectiveness

This section has been updated to reflect learning in the continued data collection for 2017. The definition for creativity has been updated to better reflect current literature and community views and the definition of inquisitiveness updated to explain the distinctions between this and life-long learning.

Personal characteristics and organizational characteristics can either enable or inhibit a systems engineer's ability to deliver value. They also impact the effects of the forces that influence the effectiveness of the systems engineer. However, it is also possible for the characteristics to be influenced by the forces, as illustrated in Figure 11.

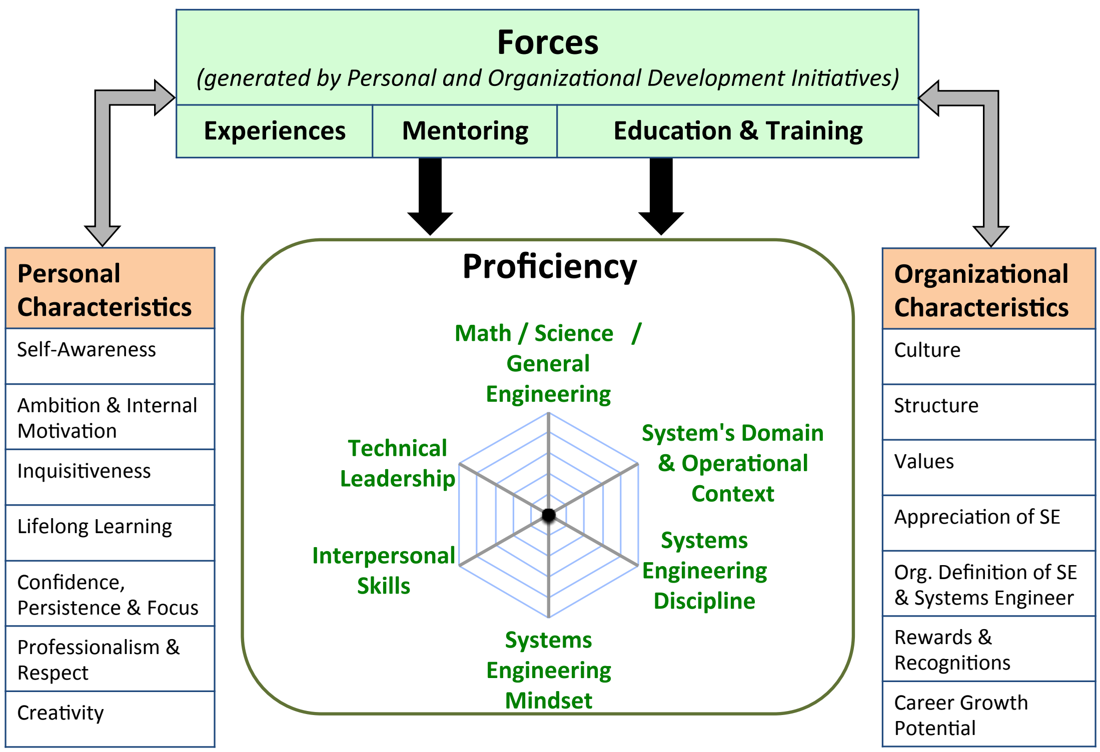

Figure 11. Forces, Proficiency and Characteristics

7.1 Personal Characteristics

Personal characteristics relate more to the personality of an individual, which implies:

- While forces that are generated through personal and organizational initiatives are expected to have a direct and significant effect on levels of proficiencies, the effect of those forces on personal characteristics is expected to be less.
- Personal characteristics are key enablers for forces to impact and grow proficiencies.

Conversely, the lack of some personal characteristics may slow down or even prevent growth of some proficiencies.

- There is not enough evidence to state whether the personal characteristics are innate or learned. However, it appears that they can be influenced or improved (examples not specific to engineering include: Freshwater 2002, Koen et al. 2012, and Coldstream 2006).

Personal characteristics tend to be a differentiator between individual systems engineers. For example, two individuals with similar educational backgrounds and experiences undergoing the same training program may accrue different levels of benefits. Significant personal characteristics, reported in the order they were most frequently described in the dataset, are:

- **Self-Awareness:** The ability to self-reflect and become aware of one's own strengths, weaknesses, knowledge, and lack thereof.
- **Ambition and Internal Motivation:** The desire to reach high career positions, and the ability to draw motivation and energy from within in order to accomplish those high ambitions.
- **Inquisitiveness**: Possessing a high level of curiosity and interest in exploring what is not yet known or understood using questions to provoke deeper or novel thinking in oneself and others.
- **Lifelong Learner:** Always looking to learn and to keeping abreast with latest developments in related disciplines and systems, irrespective of seniority or position.
- **Confidence, Persistence and Focus:** Possessing the confidence to interact with stakeholders irrespective of their relative seniority or positions; the ability to stand firm and not give-up; and the ability to remain focused on the success of the overall system.
- **Professionalism and Respect:** Being professional in the conduct, mannerisms, and ethical behaviors; and treating others with respect, recognizing that other experts may possess more knowledge and experience.
- **Creativity** Systems engineers are expected to have the ability to use their imaginations, see new possibilities in the ideas of others, find important problems, seek alternative solutions, and bring novel, useful, and valuable changes into being. Creativity is a mindset; the willingness to invent, seek, and use practical tools for innovation in the face of uncertain, ambiguous, and rapidly changing conditions.

One item to note about these personal characteristics is the relationship between inquisitiveness and lifelong learning. In *Atlas* 1.0, the definition for inquisitiveness included "hunger to keep learning" which created confusion between this and life-long learner. In reviewing the data, the real distinction between the two is that inquisitiveness may be in a specific moment or situation – curiosity that allows an individual to explore that situation fully. Lifelong learning, however, describes an individual who values continual growth and improvement over time. An individual may be inquisitive in a specific instance without having the desire for long-term growth and vice versa.

7.2 Organizational Characteristics

There are several organizational characteristics that influence how difficult or easy it may be for a systems engineer to be effective. The first grouping of characteristics is not unique to systems engineering but provides the overarching context of the organization – these characteristics would likely influence the effectiveness of any individual in the organization, regardless of her discipline, but are still critically important to understanding the context in which a systems engineer operates. The other

characteristics are specific to how an organization views, communicates about, and values systems engineering.

- **Culture, Structures, and Values:** While an organization's overarching culture, structure, and values have a much bigger impact than just on the systems engineering community, these factors certainly impact the ability of systems engineers to provide value to the organization.
    - A culture that values individual contributions over team contributions, for example, is a difficult environment for a systems engineer whose value is often realized through team coordination and interaction.
    - The way systems engineers are placed within the overall organization and how they are deployed to projects can affect performance.
    - Organizations that do state a value proposition for systems engineers tend to make systems engineering training more available and facilitate outreach with other disciplines.
- **Appreciation of Systems Engineering:** If an organization has no value proposition for systems engineers or if the value proposition for systems engineers is unclear, it raises uncertainties with individuals outside of the systems engineering community. These individuals do not understand what to expect from systems engineers or what return on investment to expect when they allocate a portion of their budget to systems engineering activities.
- **Organizational Definition of "Systems Engineering" and "Systems Engineer":** When an organization has an ambiguous definition of these terms – or no definition – it is an impediment to a systems engineer's effectiveness. In organizations lacking clear and unambiguous definitions of these terms, individuals outside of the systems engineering community form their own impression of what systems engineers do based on their personal experiences with an often limited sample of systems engineers. When the title "systems engineer" is applied loosely within an organization, it can cause tension, as people do not have clear expectations of what value a systems engineer should truly bring to a project.
- **Rewards and Recognition:** Organizations tend to have a very common and generic annual performance evaluation system; there are no specific outcomes or objectives related to the value that systems engineers provide. Organizations need a consistent means of evaluating or rewarding systems engineering practice.
- **Career Growth Potential:** In organizations where the career path for a systems engineer is obscure, the discipline is seen as less appealing than other areas where career growth and opportunity is more clearly defined.

These elements are related – for example if an organization does not define a systems engineer, it would be difficult for an individual to then understand how to progress in her career as a systems engineer and likewise it is lessens the likelihood that the organization will recognize value from systems engineering-specific efforts. This is illustrated in the example below, which reflects the Helix team's experiences with one organization.:

> *At one organization, project managers interviewed stated that when they got a "good" systems engineer, that person was critically important to helping them understand the technical vision and possibilities for a system. Good systems engineers also armed them with the information they needed to make trade-off decisions between technical capability and budget or schedule impacts. However, if they got a "bad" systems*

engineer, they were likely instead to feel encumbered with extra process – more work and restrictions – with no value added that they could define. Systems engineers in this organization stated that they were often viewed as "process wonks" because the only metrics their managers understood for systems engineers were related to formal process. They felt that if they did what they believed was good systems engineering, it was not valued. Instead the delivery of specific documents was instead used to assess their effectives. This did not align with their vision of what systems engineering should do. If the organization clearly communicated the expectations for and potential values provided by systems engineers, then managers, program managers, and systems engineers would all have a clearer understanding of effectiveness in that context. Then the organization could more clearly define and foster an appreciation for the benefit of systems engineering and reward them accordingly. This could result in an improvement of effective systems engineering, making the systems engineers feel more appreciated and rewarded for doing what they deem "the right things."

For *Atlas 1.0*, the state of organizational characteristics around systems engineering are effectively tri-modal: in the sample, organizations either show *good* practices, had *no* practices, or there was some *muddle* in between. For example, most organizations did not have any standard definition for "systems engineering" or "systems engineer" and of the organizations that did have these, there was a disconnect between the organizational view and the understanding by the systems engineers in that organization. In an organization that did have clear definitions, for instance, it was common in interviews for systems engineers to report they were hearing the "official" definitions for the first time during their interview.

8 Personal and Organizational Development Initiatives

This section is unchanged from Atlas *1.1.*

Personal development initiatives are what *individuals* do to improve their own effectiveness. Organizational initiatives are programs created by an organization with the express purpose of improving the capabilities of their systems engineering workforce. Personal initiatives do not include *participating* in organizational initiatives. For example, if an individual obtains a master's degree as a member of an organization-sponsored cohort, that would be considered an organizational initiative.

8.1 Personal Development Initiatives

When asked what personal initiatives they had for improving their own effectiveness, 100% of the systems engineers in the sample participated in organizational initiatives in some ways – most specifically in mandatory training or mentoring programs. Many fewer individuals had personal growth initiatives (7%) outside of the initiatives of their organizations. There were a few common approaches:

- *Individual Reading* – Some individuals reported that they spent personal time reading material related to their work; e.g., journal articles, conference papers, trade publications, relevant news or magazine articles, or books. Journal articles, conference papers, trade publications, and new articles tended to be around technical subjects – new technologies related to the systems the individual supported, classic engineering disciplines, relevant domains, or systems engineering itself (such as the INCOSE *Systems Engineering* journal or the IEEE *Systems* journal). When individuals read books for self-development, they were more commonly on non-technical topics such as technical leadership - particularly business – or interpersonal skills – particularly communication.
- *Attending conferences* – Several individuals stated that they attended conferences relevant to their work whenever possible – generally, a mix of domain-specific, classic engineering, systems engineering, or project management conferences. Individuals who attended conferences stated that their organizations sponsored their attendance, but that this was not a broad initiative; rather, their individual managers or programs helped them find funding to attend relevant events. A few individuals said that they used to attend conferences, but that funding was no longer available for these efforts and had not been for the last five years or more.
- *Online courses* – these are not full academic courses for credit that could be counted towards a degree. Those types of courses were considered education. However, a few individuals indicated that there were free courses available online; e.g., massive open online courses (MOOCs) or small, university-sponsored free courses on relevant topics. Popular topics included overviews of basic classic engineering disciplines such as electrical or software engineering, as well as risk- or decision-management, and specific technology areas. Individuals who took these courses said they were helpful to master an overview of an area, particularly on topics that were relevant to the systems on which an individual worked, but in which she did not have experience. Because these courses are not sponsored by the company, taking them is wholly dependent on individual motivation.
- *Certification – All* DoD organizations required an engineering certification (at the time of the Helix interviews, the Systems Planning, Research, Development, and Engineering (SPRDE)

certification) for all of their systems engineers. However, a few individuals had also sought additional certification. No organization specifically sponsored external certification initiatives, and the few individuals who had become certified said that they did not believe that it would help them in their organizations. They felt additional certification was important for them as individuals. The three types of certifications discussed were INCOSE Certified Systems Engineering Profession (CSEP); PMI Project Management Profession (PMP); and state-certified Professional Engineer (PE). Note that only the first certification is unique to systems engineering.

Of the individuals who stated they did not do anything outside of organizational initiatives, many junior and mid-level systems engineers said that they would like to, but that there are roadblocks. The most commonly stated are time-consuming work responsibilities and managers who do not support additional training. In one organization, individuals stated that they were expected to pursue training but were not given leave from their roles and were "dinged on their performance" for failing to get additional training. Most senior systems engineers who discussed personal initiatives stated that beyond reading or attending conferences, they believed building on their experiences was sufficient. However, almost 5% of senior systems engineers had at one point created training programs specifically to pass on their knowledge and experiences to younger systems engineers in their organizations.

8.2 Organizational Development Initiatives

Helix identifies 'initiatives' (both personal and organizational), as those that are intended to generate one or more the forces (experiences, mentoring, and education & training) in a direct manner. These forces, in turn, are expected to improve the proficiency of an individual systems engineer. This section presents various aspects of organizational development initiatives that were discussed during Helix interviews, with a particular focus on initiatives that are available for the benefit of the systems engineers in the organization.

The discussion presented in this section is aggregated from the 40% of all Helix interviews in which participants discussed organizational initiatives. In organizations with a larger number of Helix participants, a richer view of the organization emerged, sometimes with conflicting views presented by the participants. While these are highlighted in the discussion, the intent is not to provide an organization level analysis of initiatives.

8.2.1 Nature of Organizational Initiatives

Many features of organizational characteristics can be observed from Helix interviews:

- **Distinction between initiatives and policies**: It is not always straightforward to recognize and identify organizational initiatives, and to distinguish them from organizational practices and policies. Helix considers it an initiative if the organization plays an active role in promoting, enabling, and supporting it for the benefit of its employees. For example:
  - Some organizations provide tuition reimbursement to their employees seeking graduate degrees in related disciplines, subject to policies regarding eligibility, absence from work, etc. Typically, it is up to the individual employee and her immediate supervisor to take advantage of those policies.

  - Other organizations play a more active role in providing graduate education for their employees: they establish relations with specific universities; they establish cohorts for individual courses and/or degree programs; they provide facilities within their premises for the universities to conduct courses; they make available organizational data for projects and dissertations; and also tend to reward employees who go through these programs with a promotion or salary raise.

- **Scope of organizational initiatives**: Some organizational initiatives are targeted at systems engineers' proficiencies, systems engineering proficiencies of the workforce, or within the systems engineering department/division. There are initiatives that are offered only to those systems engineers that meet certain eligibility criteria and not to the entire systems engineering population. These "high potential" programs are generally intended to help selected systems engineers mature more rapidly. There are also other initiatives intended for the benefit of all employees across the entire organization, which include any systems engineers; for example, some organizations will pay for any graduate education, regardless of subject. Each of these can be a benefit to a systems engineer, though programs scoped specifically to the systems engineering population tend to be more directly beneficial.

- **Influence of organizational initiatives on organizational characteristics:** While some organizational initiatives generate forces that in turn improve the proficiency levels of individual systems engineers, some other organizational initiatives improve organizational characteristics – either directly or indirectly. For example:

  - Some organizations have initiatives to identify and recruit SE talent from within the organization, and also to recognize and reward achievements of systems engineers and other employees. Such initiatives do not directly improve any of the forces, but rather the organizational characteristics.

  - Some organizations have mentoring initiatives to develop their junior systems engineers by pairing them up with senior systems engineers. Such initiatives are intended to directly benefit the mentee. However, such relationships between junior and senior systems engineers also tend to improve the environment and culture of the organization. (See Section 6.2.4 on the benefits of mentoring.)

- **Formal and informal initiatives**: By definition, organizational initiatives are formally established and deployed. However, there are also informal versions of those formal initiatives that could even co-exist with formal versions within the same organization. Some informal initiatives are also established by the organization. For example:

  - It is typical for mentors and mentees to form an informal mentoring relationship, without being explicitly directed by the organization. Such informal mentoring relationships tend to exist irrespective of the establishment of a formal mentoring initiative in that organization.

  - Some organizations offer a variety of training courses on topics of relevance, often in a classroom setting. In addition, there are also informal training and information sessions that the organization offers – as guest lectures or lunch-and-learn programs.

- **Portfolio of initiatives**: Organizational initiatives rarely exist in isolation; typically, a portfolio of initiatives is available to employees. Organizations establish individual initiatives to address various needs; and in some cases, a higher-level initiative leads to many lower level initiatives as well. For example, an organization may have mentoring and rotational programs. These may be

linked, such that each new rotation pairs an individual with a new/additional mentor. An individual in the rotation program, then, not only gains skills from new work experiences, but also develops a larger network of trusted individuals on whom she can call for advice and support.

As another example, an organization may have a goal to increase the percentage of the workforce with graduate degrees and creates an incentive program for graduate education, paying for tuition and giving an individual a number of paid hours each week to devote to study. If many systems engineers take advantage of this to gain formal systems engineering education and the organization identifies clear positive impacts, the organization may decide to partner with a university to develop a cohort program for systems engineering master's education.

8.2.2 Types of Organizational Initiatives

Participants in Helix interviews discussed the features, benefits, and shortcomings of many organizational initiatives that they had either directly participated in or have been aware of – both in their current organizations and in their previous organizations. The many initiatives mentioned, may be classified under the following types:

- **Recruitment initiatives**: These initiatives recognize systems engineering talent and bring individuals into the systems engineering fold. In some organizations, such initiatives bring in new employees from outside the organization – usually fresh graduates or others with limited experience. Other organizations have initiatives to recognize and recruit systems engineers from elsewhere in the organization, usually after a manager has identified the person as a "systems thinker".
- **Orientation initiatives**: Some initiatives are exclusively targeted at new employees to familiarize them with the organization, its processes, and the way it does systems engineering. In most organizations, a job rotation program is usually offered only to new / junior employees, offering them a glimpse into various parts of the organization before assigning them to one part of the organization. Some organizations recognize the value of such initiatives to senior employees, and extend those initiatives to them as well.
- **Experience enhancing initiatives:** Junior systems engineers grow into senior experienced systems engineers not just by the number of years they spend in an organization, but through performing in various systems engineering roles; different projects; various levels and types of systems; and different phases of a systems lifecycle. Organizations establish initiatives that are designed to effectively provide rich experiences to systems engineers. Typically, these take the form of rotational programs with specific paths depending on the types of skills to be developed.
- **Mentoring initiatives**: These initiatives are very prevalent in many organizations – either as a formal or an informal arrangement. While the primary beneficiaries of mentoring arrangements are the less experienced mentees, the more experienced mentors and the organization at large stands to benefit as well. From a Helix perspective, 'mentoring' is also identified as a force that directly impacts and enhances the proficiency of systems engineers. Section 6.2 provides additional discussion on mentoring and mentoring initiatives.
- **Education and training initiatives:** Every employee enters any organization with some level of

formal education. Recognizing the value of formal education, many organizations offer many initiatives for their employees to obtain higher degrees from universities. There is also a need for employees to be trained in particular specialized topics, and organizations typically offer many training options of varying types and durations for the benefit of its employees. Various aspects of training are discussed in Section 6.3.

- **Knowledge management initiatives**: A significant risk in many of the organizations that participated in the Helix interviews was the imminent loss of senior system engineers and their vast experiences. Many organizations have established initiatives to capture those experiences in various ways, and to store them in a readily accessible manner as when required.
- **Leadership development initiatives:** The most senior technical position that a systems engineer can achieve in an organization is that of a chief systems engineer or equivalent. Organizations tend to identify high-potential employees from its pool of junior and mid-level systems engineers, and offer them initiatives to enhance their leadership proficiencies in addition to technical proficiencies, thus enabling those systems engineers to develop in to future chief systems engineers and other senior systems engineering positions.
- **Rewards and recognition initiatives:** As a way to motivate, encourage, and appreciate the achievements of its systems engineers, organizations establish various rewards and recognition initiatives specifically for systems engineers in addition to its employees at large.

Overall, initiatives are focused on helping individuals develop additional proficiency using one or more of the forces identified in *Atlas*. For example, rotational programs are designed to increase the breadth of experiences. Apprentice programs – where an individual is paired with a more senior individual and shadows them – provides an opportunity for building proficiencies through both experiences and mentoring. Rewards initiatives generally help to identify and provide solid examples of effective systems engineers, highlighting the key systems engineering values for the organization.

8.2.3 Phases of Organizational Initiatives

Helix interview data indicates that organizational initiatives tend to have various phases. Appropriate recognition and management of initiatives across these different phases is critical for success.

- **Identifying the need:** The first step in any organizational initiative is to clearly articulate the need for one, or define the problem that needs to be solved. While there are many types of initiatives that an organization could potentially establish, it is imperative for an organization to understand why a particular initiative is required.
- **Establishing the initiative**: Once the need is recognized and the type of initiative is identified, the organization must then establish the initiative by setting up the required policies, guidance, personnel to run / manage the initiative, criteria for selecting beneficiaries, and the required infrastructure.
- **Deploying the initiative:** There are a number of activities to be done once the organization has established an initiative:
    - *Promoting:* In 90% of the organizations that participated in Helix interviews, there were initiatives that were wholly unknown to at least one Helix interviewee. The organization must take an effort to let its employees know of any initiative that they can benefit

from. Newer employees who go through some sort of an orientation tend to be more aware of initiatives that they can immediately benefit from. Even those employees who have spent many years in the organization are not very aware of the initiatives that are available to them.

  - *Enabling*: When an employee is interested in a particular initiative and is eligible, the organization must enable the employee to benefit from that initiative. Experiences shared by Helix participants indicate that there are situations when they are unable to take advantage of an organizational initiative since they could not take time off their regular work to participate in a training initiative, or that some procedures diminished the effectiveness of the initiative.

- **Responding to outcomes of initiatives**: When an employee participates and benefits from an initiative, typically, there are new skills or knowledge that are acquired, and the employee could recommend improvements based on this. For example, if an employee receives education or training on systems engineering processes, and if the organization does not support modification of existing systems engineering processes, it defeats the purpose of the education.

- **Evaluating the initiative:** The most critical aspect of the success of an initiative is to evaluate it periodically, and to then update, reform, stop, or restart an organizational initiative. A critical evaluation could also reveal enablers and inhibitors for the initiatives. Helix interviews indicated evidence of many situations:
  - Initiatives no longer address the need for which they were established.
  - The need for which an initiative was established is no longer valid.
  - There are more trainers than trainees.
  - Employees are not motivated.
  - The evaluation of some initiatives makes it appear more successful than it really is.
  - The procedures and policies for an initiative could be burdensome.
  - There is a need to restart an initiative that used to be very effective but was stopped due to many reasons, including budget cuts.
  - The duration of a training course may be altered.
  - The target beneficiaries for an initiative need to be redefined.

9 CONCLUSIONS

This presentation of *Atlas* is intended to present all of the critical aspects of the Theory of Effective Systems Engineers. It provides an overview of all elements of *Atlas* as well as enough details to be used by individuals and organizations. However, the team strongly recommends than an individual or organization also reference the following companion documents:

- **Atlas *1.1 Implementation Guide: Moving from Theory into Practice*** – This document provides detailed guidance for individuals and organizations looking to use Atlas for growth and development. It was generated based on the Helix team's experiences working with 22 organizations in the Helix dataset as well as helping several organizations think though how to use *Atlas*.

- ***Atlas Career Path Guidebook*** – This document provides analyses of the Helix dataset, providing common patterns in systems engineers' careers. The Guidebook also provides some insights on questions commonly asked of the Helix team around career paths and the team's responses. Finally, additional work on linking proficiencies to career paths has been completed and is reflected in the guide. (SERC-2018-TR-101-C)

- ***2017 Helix Technical Report*** – This document provides an overview of the work completed in 2017 along with the team's vision and planning for future Helix work. It references, rather than repeats, the findings of the other documents. In addition, it captures the detailed methodologies utilized on the Helix project. (SERC-2018-TR-101)

Each of these documents can be found at sercuarc.org/projects/Helix.

References

Bellinger, G. 2004. “Modeling and Simulation: An Introduction.” Webpage. Accessed 30 August 2014. Available at: http://www.systems-thinking.org/modsim/modsim.htm

BKCASE Editorial Board. 2016. *The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 1.7*. R.D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed 27 October 2016. www.sebokwiki.org.

Cameron, K. and R.E. Quinn. 2011. *Diagnosing and Changing Organizational Culture: Based on the Competing Values Framework*: Based on the Competing Values Framework, 3rd Edition. Hoboken, NJ: Wiley.

Coldstream, P. 2006. “Training Minds for Tomorrow’s World: on Independence, Curiosity, and Effectiveness.” *Higher Education in Europe*. 22(4): 507-515.

Harris, G. and K. Eikenberry. 2016. “DISC Profile and Human Behavioral Assessments Provider: What is the DISC Model?” https://discpersonalitytesting.com/blog/what-is-the-disc-model/

DoD. 2013. *Interim DoD Instruction 5000.2: Operation of the Defense Acquisition System*. Arlington, VA: U.S. Department of Defense. 26 November 2013.

Freshwater, D. (ed.). 2002. *Therapeutic Nursing: Improving Patient Care through Self Awareness and Reflection.* London, England: Sage Publications Ltd.

Harrison, R. 1972. “Understanding your organisation's character.” Harvard Business Review.

Hutchison, N. *A Framework to Classify Experiences and Enable Career Path Analysis to Support Maturation of Effective Systems Engineers in the Defense Industry*. PhD Dissertation. Hoboken, NJ: Stevens Institute of Technology. October 2015.

Hutchison, N., A Pyster, D. Henry. 2015. “Trends in the Education of Systems Engineers.” Submitted to *Systems Engineering* March 2015.

Hutchison, N., A. Pyster, D. Henry, C. Barboza. 2015. “Understanding What Makes Systems Engineers Effective.” Submitted to *Systems Engineering* March 2015.

INCOSE. 2014. *A World in Motion: Systems Engineering Vision 2025*. San Diego, CA, USA: International Council on Systems Engineering.

INCOSE UK. 2010. "Systems Engineering Competency Framework," Accessed on June 3, 2015. Available at, http://www.incoseonline.org.uk/Normal_Files/Publications/Framework.aspx?CatID=Publications&SubCat=INCOSEPublications.

ISO/IEC. 2009. "Systems and Software Engineering Vocabulary (SEVocab)" - ISO/IEC 24765. in International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) [database online]. Geneva, Switzerland, 2009 [cited December 21 2009]. Available at: http://pascal.computer.org/sev_display/index.action.

ISO/IEC/IEEE. 2008. *Systems and Software Engineering -- System Lifecycle Processes.* Geneva, Switzerland: International Organisation for Standardisation/International Electrotechnical Commissions. ISO/IEC/IEEE 15288:2008.

ISO/IEC/IEEE. 2010. *Systems and Software Engineering - System and Software Engineering Vocabulary (SEVocab)*. Geneva, Switzerland: International Organization for Standardization

(ISO)/International Electrotechnical Commission (IEC)/ Institute of Electrical and Electronics Engineers (IEEE). ISO/IEC/IEEE 24765:2010.

Koen, J., U-C. Klehe, and A.E.M Van Vianen. 2012. “Training Career Adaptability to Facilitate a Successful School-to-Work Transition.” *Journal of Vocational Behavior*. 81(3): 395-408. December 2012.

Maximini, D. 2015. *The Scrum Culture: Introducing Agile Methods in Organizations*. Cham, Switzerland: Springer.

Pyster, A., D.H. Olwell, T.L.J. Ferris, N. Hutchison, S. Enck, J. Anthony, and D. Henry (eds.). 2015. *Graduate Reference Curriculum for Systems Engineering (GRCSE®) v.1.1*. Hoboken, NJ, USA: Trustees of the Stevens Institute of Technology.

Pyster, A., P. Dominick, D. Henry, N. Hutchison, C. Lipizzi, M. Kamil, S. Manchanda. 2014(b). Atlas*: The Theory of Effective Systems Engineers*, v. 0.25. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-4.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2013. *Helix – Phases 1 and 2.* Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2013-TR-038-2.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2014. *Helix – Phase 3*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-3.

Sheard, S. 1996. “Twelve Systems Engineering Roles.” Proceedings of the International Council on Systems Engineering (INCOSE) Sixth Annual International Symposium, Boston, Massachusetts, USA, 1996.

Sheard, S. 2000. “Systems Engineering Roles Revisited.” Software Productivity Consortium, NFP.

APPENDIX A: LIST OF HELIX PUBLICATIONS

Capstone Project

Partacz, M. 2017." Building a Better Business Case for Systems Engineering: The Relationship between a Systems Engineer's Career Path, Proficiency and Project Performance." Capstone Project Report. Hoboken, NJ: Stevens Institute of Technology, Hoboken, NJ.

Conference Papers and Presentations

Clifford, M. and N. Hutchison. 2017. "Helix: Understanding Systems Engineering Effectiveness through Modeling." Proceedings of the National Defense Industrial Association (NDIA) Systems Engineering Conference, 25 October 2017, Washington, DC.

Henry, D., N. Hutchison, A. Pyster, P. Dominick, C. Lipizzi, M. Kamil, S. Manchanda. 2014. "Summary of Findings from the Helix Project (2013-14) - An Investigation of the DNA of the Systems Engineering Workforce". Proceedings of the National Defense Industrial Association (NDIA) Systems Engineering Conference, Springfield, VA, USA, October 23-26, 2014.

Hutchison, N., D. Henry, A. Pyster, and P. Dominick. 2014. "Systems Engineering Career Analysis: Supporting a Theory of Systems Engineers." Proceedings of the INCOSE Europe, Middle-East, Africa Systems Engineering Conference (EMEASEC), Cape Town, South Africa, October 27-30, 2014.

Hutchison, N., D. Henry, A. Pyster, P. Dominick. 2014(b). "Systems Engineering Career Analysis: Supporting a Theory of Systems Engineers." Proceedings of the Europe, Middle East, and Asia Sector Systems Engineering Conference, 27-30 October 2014, Cape Town, South Africa.

Hutchison, N., Henry, D., Pyster, A., Pineda, R. 2014. "Early Findings from Interviewing Systems Engineers who Support the US Department of Defense." Proceedings of the 2014 International Council on Systems Engineering (INCOSE) 24th International Symposium, June 30-July 3, 2014, Las Vegas, NV.

Hutchison, N., J. Wade, and S. Luna. 2017, "The Roles of Systems Engineers Revisited." Proceedings of the International Council on Systems Engineering (INCOSE) 27th International Symposium, 15-20 July 2017, Adelaide, Australia.

Jauregui, C., A. Pyster, D. Henry, N. Hutchison, and C. Wright. 2016. "Insights on the Experiences and Education of INCOSE-Certified Expert Systems Engineering Professionals and Chief Systems Engineers." Proceedings of the International Council on Systems Engineering (INCOSE) 26th International Symposium, 18-21 July 2016, Edinburgh, Scotland.

Lipizzi, C., S. Manchanda, M. Kamil, A. Pyster, D. Henry, N. Hutchison. 2015. "The Education Background of INCOSE Systems Engineering Professional Certification Program Applicants." Proceedings of the International Council on Systems Engineering (INCOSE) International Symposium, Seattle, WA, July 2015.

Pyster, A. and N. Hutchison. 2015. “The Helix Project: Analysis of INCOSE SE Certification Program Applicants.” Special Presentation to the International Council on Systems Engineering (INCOSE) Corporate Advisory Board (CAB), January 25, 2015 as part of the INCOSE International Workshop, 25-27 January 2015, Los Angeles, CA.

Pyster, A., N. Hutchison, D. Henry, C. Barboza. 2015. “Helix: What Makes Systems Engineers Effective?” Proceedings of the Conference on Systems Engineering Research (CSER), March 17-19, 2015, Hoboken, NJ.

Pyster, A., R. Pineda, D. Henry, N. Hutchison. 2013. “Helix: Investigating the DNA of the Systems Engineering Workforce.” Proceedings of the National Defense Industrial Association (NDIA) Systems Engineering Conference, October 28-31, 2013, Crystal City, VA.

Pyster, A., S. Rifkin, D. Henry, K. Lasfser, N. Hutchison, D. Gelosh. 2013. "The Helix Project: Understanding the Systems Engineers who Support the US Department of Defense." Proceedings of the Asia-Pacific Conference on Systems Engineering (APCOSE), 8-11 September 2013, Yokahama, Japan.

Squires, A., Wade, J., Hutchison, N. 2016. “Building a Pathway to Systems Education for the Global Engineer.” Proceedings of the American Society for Engineering Education (ASEE) 123rd Annual Conference, 26-29 June 2016, New Orleans, Louisiana, USA.

<u>Dissertation</u>

Hutchison, N. *A Framework to Classify Experiences and Enable Career Path Analysis to Support Maturation of Effective Systems Engineers in the Defense Industry*. PhD Dissertation. Hoboken, NJ: Stevens Institute of Technology. October 2015.

<u>Journal Article</u>

Hutchison, N., A. Pyster, D. Henry. 2016. "Atlas: Understanding What Makes Systems Engineers Effective in the US Defense Community." *Systems Engineering*. 19(6): 510-521.

<u>Project Reports</u>

Hutchison, N., D. Henry, D. Verma, M. Clifford, A. Pyster. 2016. *Atlas: The Theory of Effective Systems Engineers, version 1.0*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. 16 December 2016.

Hutchison, N., D. Henry, D. Verma, M. Clifford, R. Giffin, A. Pyster. 2016. *The Development of* Atlas*: The Theory of Effective Systems Engineers*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2016-TR-118. 16 December, 2016.

Hutchison, N., D. Verma, P. Burke, M. Clifford, R. Giffin, S. Luna, M. Partacz. Atlas *Career Path Guidebook: Patterns and Common Practices in Systems Engineers' Development*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2018-TR-101-C. 16 January 2018.

Hutchison, N., D. Verma, P. Burke, M. Clifford, R. Giffin, S. Luna, M. Partacz. Atlas *1.1: An Updated to the Theory of Effective Systems Engineers*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2018-TR-101-A. 16 January 2018.

Hutchison, N., D. Verma, P. Burke, M. Clifford, R. Giffin, S. Luna, M. Partacz. Atlas *1.1 Implementation Guide: Moving from Theory into Practice*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2018-TR-101-B. 16 January 2018.

Hutchison, N., D. Verma, P. Burke, M. Clifford, R. Giffin, S. Luna, M. Partacz. *RT-173: Helix – 2017 Helix Technical Report*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2018-TR-101. 16 January 2018.

Pyster, A., D. Henry, N. Hutchison, C. Jauregui, M. Clifford. 2015. *Atlas: The Theory of Effective Systems Engineers, version 0.5*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2015-TR-108. 1 December 2015.

Pyster, A., N. Hutchison, D. Henry, C. Jauregui, M. Clifford. 2015. *Extending* Atlas *to Non-Defense Systems Engineers and to Classic Engineers*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2015-TR-110. 18 December 2015.

Pyster, A., P. Dominick, D. Henry, N. Hutchison, C. Lipizzi, M. Kamil, S. Manchanda. 2014(b). Atlas*: The Theory of Effective Systems Engineers*, v. 0.25. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-4. 21 November 2014.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2013. *Helix – Phases 1 and 2.* Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2013-TR-038-2.

Pyster, A., R.L. Pineda, D. Henry, N. Hutchison. 2014. *Helix – Phase 3*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology. SERC-2014-TR-038-3.

Workshop Reports

Hutchison, N., D. Verma, P. Burke, M. Clifford, S. Luna, M. Partacz, R. Giffin. *Report on the 4th Annual Helix Workshop*. 4th Annual Helix Workshop, 17 October 2017, MITRE Campus, McLean, Virginia.

Hutchison, N., D. Verma, R. Giffin, M. Clifford, A. Pyster. 2016. *Report on the Helix Early Adopter's Workshop*. Helix Early Adopter's Workshop, 20 September 2016, Ronald Reagan International Trade Center, Washington, DC.

Pyster, A., D. Henry, N. Hutchison, C. Jauregui, J. Armstrong, M. Clifford. 2015. *Report on the Second Helix Workshop: Exploring the Theory of Systems Engineers' Effectiveness*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology.

Pyster, A., P. Dominick, D. Henry, N. Hutchison, C. Lipizzi, M. Kamil. 2014(a). *Report on the First Helix Workshop: Exploring the Theory of Systems Engineers' Effectiveness*. Hoboken, NJ: Systems Engineering Research Center, Stevens Institute of Technology.

In process

A paper on systems engineering career paths, "Discovering Career Patterns in Systems Engineering" has been submitted for the 2018 INCOSE International Symposium. The authors are Nicole Hutchison, Sergio Luna, and Matthew Partacz.

Other

ABET Symposium 2016, Fort Lauderdale, FL – ABET panel on systems engineering education and research for the 2016 ABET conference. Nicole Hutchison presented on Helix.

INCOSE Healthcare Systems Engineering Working Group Webinar – November 29, 2016. Nicole Hutchison delivered a webinar, a 60-minute overview of Atlas with specific implications related to healthcare systems engineers.

Atlas Self-Assessment Tool. An Excel-based tool published 16 December 2016. Available at http://sercuarc.org/projects/Helix

Helix Team. 2016. *Guide to* Atlas *1.0 Self-Assessment Tools.* A companion users guide for the *Atlas Self-Assessment Tool* published 16 December 2016. Available at http://sercuarc.org/projects/Helix

Appendix B: Glossary and Terminology

Consistency in the definition and understanding of terminology and concepts is essential for any deliberation. This section presents the definitions and classifications that are relevant to *Atlas*. Some have been obtained from available literature, while others have been created specifically for *Atlas*.

Acronyms and Abbreviations

CSE	Chief Systems Engineer
DASD(SE)	U.S. Deputy Assistant Secretary of Defense for Systems Engineering
DIB	Defense Industrial Base (supports DoD)
DoD	U.S. Department of Defense
GRCSE	Graduate Reference Curriculum for Systems Engineering
INCOSE	International Council on Systems Engineering
IR&D	Internal (or Independent) Research & Development
MBA	Master of Business Administration
SE	Systems Engineering
SERC	Systems Engineering Research Center
SEBoK	*Guide to the Systems Engineering Body of Knowledge*
SME	Subject Matter Expert
UARC	University-Affiliated Research Center

Atlas Definitions

- **Systems Engineer**

  > *A **Systems Enginee**r is an individual who performs systems engineering activities and is recognized (either formally or informally) by his or her organization for her ability to perform these activities.*

  This definition of a systems engineer does *not* refer to the title that someone may hold in her organization. Someone may never hold the title 'Systems Engineer', but could be considered to be one based on the activities she performs. Similarly, someone may hold the title 'Systems Engineer', but her activities may not be considered to be systems engineering activities.

- **Effective Systems Engineer**

  > *An **Effective Systems Engineer** is someone who consistently delivers value by performing systems engineering activities in positions assigned by the organization.*

  This definition is fundamental to *Atlas* since the focus of Helix research is the effectiveness of systems engineers. Though 'effectiveness' is a subjective term, this definition ties it to 'value' that can be defined and even measured – qualitatively, if not quantitatively.

- **Chief Systems Engineer (CSE)**

*A **Chief Systems Engineer (CSE)** is one who has formal responsibility to oversee and shepherd the technical correctness and to maintain a consistent vision for a system, often coordinating with many other systems engineers who have smaller scopes of responsibility.*

The Chief Systems Engineer (CSE) position is one of the most senior technical positions that system engineers can achieve while staying in a technical track (as opposed to a management track). Though the title 'Chief Systems Engineer' is not used in all organizations, the concept of a CSE position (or equivalent) is common, especially in industry. There is no consistent description of a CSE's (or equivalent's) formal authority, but overall responsibility for a system is often split in some way between the CSE and the project or program manager (PM).

- **Position**

*A **Position** held by an individual is equivalent to a 'title', where the organization defines what roles and responsibilities it entails.*

This definition of a position is usually specific to an organization and does not translate across organizations.

- **Role**

*A **Role** performed by an individual consists of a specific set of related activities.*

Typically, an individual performs multiple roles in any given position. In the context of *Atlas*, the roles of interest are systems engineering roles.

- **Career Path**

*An individual's **Career Path** is the precise combination (in terms of characteristics, timing, and order) of experiences, mentoring, and education and training that they undergo during their entire career.*

This definition, created for *Atlas*, is different from how career paths are typically defined in the human resources (HR) community. HR definitions tend to be focused on rigid hierarchy that may be useful for HR classification and management of positions within an organization. However, they provide little insight into the growth and development of individuals throughout their career, particularly across organizations.

- **Proficiency**

*The **Proficiency** of an individual is the quality or state of knowledge, skills, abilities, behaviors, and cognition.*

In *Atlas*, the term 'proficiency' is used broadly to include everything that an individual needs to be good at in order to be an effective systems engineer. This distinguishes *Atlas* from competency models that tend to focus primarily on the discipline of systems engineering.

- **Seniority of a Systems Engineer**

  As systems engineers traverse the path of their careers from the point of entry into the workforce (or recruitment) to the point or exit from the workforce (or retirement), there is a continual maturation that is reflected in the breadth and depth of their proficiencies; the types of roles & positions they play; and the value that they provide or that is expected from them. Grouping systems engineers under some levels of 'seniority' that reflect the levels of maturation enables patterns to be identified across systems engineers, and insights to be drawn from them.

  Helix has identified three levels of seniority in systems engineers: junior, mid-level, and senior. Traditionally, 'number of years of work experience' has been used as a preliminary criterion for distinguishing between these levels of seniority, but it fails to capture the nuances of differentiation within systems engineers. Hence, it is not included in Table 10 that states various criteria used to distinguish between junior, mid-level, and senior systems engineers. These criteria are meant to be indicative and not rigid; there are always examples of specific individuals whose seniority is not consistent with these criteria.

Table 10. Criteria for Distinguishing the Seniority of Systems Engineers

Criteria for Distinguishing the Seniority of Systems Engineers			
Criteria	**Junior**	**Mid-level**	**Senior**
Leadership	Primarily works as an individual contributor; has had zero or one formal leadership positions, which can be as an official supervisor or as a task leader	Has had at least two formal leadership positions over teams or tasks of significant size and scope; viewed as a leader in a project, program, or business unit of the larger enterprise	Three or more formal leadership positions over teams or tasks of significant size and scope, including second-level management roles; viewed as a leader in the enterprise
Complexity	Relevant experiences on a simple project, system, or task, working primarily at the system components level or simple activities such as managing a requirements database	Relevant experiences on moderately complex projects or systems, working at the sub-system and system levels or on moderately complex activities such as managing the development and negotiation of requirements for a moderately complex system	Relevant experiences on complex projects or systems, working at the system and platforms/systems of systems levels or on quite complex activities such as managing the development and negotiation of requirements for a complex system of systems
Lifecycle	Relevant experiences in	Relevant experiences in	Relevant experiences in

W i	at least two phases of the systems lifecycle	at least three phases of the systems lifecycle	at least four phases of the systems lifecycle
t h **Roles**	Worked on up to 3 different roles	Worked on 4 to 6 different roles	Worked on 7 to 15 different roles

With respect to Table 10:

1. Experience is considered to be 'relevant' if it directly supports the growth of systems engineering proficiencies.
2. A leadership position is 'formal' if it is officially defined and recognized by the organization. This does not mean that the individual necessarily has organizational authority over the individuals she is leading. Likewise, there is no defined minimal team size. Typically, early leadership positions are over small teams (less than five people) and as the individual matures, the size of the teams increases.
3. The hierarchy of system levels (components -> subsystems -> systems -> system of systems) is based on definitions from the *Guide to the Systems Engineering Body of Knowledge* (BKCASE Editorial Board 2016) and reflects system complexity and completeness, where 'parts' at any level are combined to form the 'whole' at the next level.
4. The various aspects of the systems lifecycle are based on definitions from the *Guide to the Systems Engineering Body of Knowledge* (BKCASE Editorial Board 2017) and are elaborated in Section 6.1.
5. The roles are based on the 15 systems engineering roles defined in *Atlas* 1.1.

Formal education, titles, and roles are *not* considered to be distinguishing criteria, since they cannot be used to consistently draw any distinctions between levels of seniority of systems engineers. However, as a baseline, systems engineers typically have an undergraduate degree in a STEM (science, technology, engineering, and mathematics) field.